General Supervised Learning Framework for Open World Classification

SAI KRISHNA THEJA BHAVARAJU, School of Industrial and Systems Engineering, University of Oklahoma, USA
MOHAMMAD AMIN BASIRI, Data Science and Analytics Institute, University of Oklahoma, USA
CHARLES NICHOLSON*, School of Industrial and Systems Engineering, University of Oklahoma, USA

In open-world supervised learning for classification, the training data is incomplete with respect to the full set of relevant classes in the application domain. Most existing research on this problem focuses on computer vision, and many of the proposed methodologies are intrinsically tied to specific machine learning algorithms or data types. However, real-world open-world settings may arise in a wide array of problem contexts, each with its own data type and classifier requirements. Although existing research emphasizes the identification of unknown sets or classes, it does not sufficiently address automatically categorizing these new classes and updating predictive models. In this work, we present a framework that addresses all aspects of the open world classification pipeline. The proposed approach is data- and model-agnostic, making it versatile across different domains. Our framework performs automatic identification and categorization of unknown instances into distinct new classes while dynamically updating predictive models without human intervention. We evaluate it on diverse data types, including images, text, and sensor data, demonstrating effectiveness across experiments with accuracy improvements ranging from 27 to 69 percentage points. To assess robustness and provide practical guidance, we conduct comprehensive sensitivity analysis examining the impact of key parameters including the number of known classes, the Chebyshev confidence parameter, the itemset size parameter, and base classifier quality. Additionally, we provide insights into practical applications through a case study on social media analytics for disaster response, highlighting the adaptability of the framework in real-world scenarios.

JAIR Associate Editor: Prof. Chang-Dong Wang

JAIR Reference Format:
Sai Krishna Theja Bhavaraju, Mohammad Amin Basiri, and Charles Nicholson. 2026. General Supervised Learning Framework for Open World Classification. *Journal of Artificial Intelligence Research* 85, Article 18 (February 2026), 32 pages. DOI: 10.1613/jair.1.20947

1 Introduction

In traditional supervised learning for classification, models are trained based on datasets that contain examples of all classes to be identified. That is, in so-called closed-world problems, all classes are known in advance. Once the model is trained, it can be used to predict or otherwise discriminate between these same classes in new data. However, there are problems where this condition does not hold. In open-world problems, the training data is incomplete, and the new data for which the model is developed may contain classes that the model was not trained on.

*Corresponding Author.

Authors' Contact Information: Sai Krishna Theja Bhavaraju, krishna.theja98@gmail.com, School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK, USA; Mohammad Amin Basiri, ORCID: 0000-0002-2005-0393, ma.basiri@ou.edu, Data Science and Analytics Institute, University of Oklahoma, Norman, OK, USA; Charles Nicholson, ORCID: 0000-0002-7023-8802, cnicholson@ou.edu, School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK, USA.

DOI: 10.1613/jair.1.20947

This problem has its origins in the field of computer vision for recognition, in which a target image class (or set of classes) should be recognized among known and unknown image classes. Indeed, much of the research on open-world problems is concentrated within the domain of computer vision and referred to as open set recognition (OSR) or open-world recognition (OWR) (Bendale and T. Boult 2015; Geng, S.-J. Huang, et al. 2021; Scheirer, Rezende Rocha, et al. 2013). In one of the earliest works in OSR, the authors formally defined terms such as open space and openness that correspond to the open set recognition problem (Bendale and T. Boult 2015; Geng, S.-J. Huang, et al. 2021), and also showed how this problem setting differs from general data modeling tasks. Later works extended the concept of OSR to OWR, identifying the necessary tasks for an effective modeling system, i.e., the system should be able to detect unknown classes, label unknown points, and update the model (Han et al. 2021; Scheirer, Rezende Rocha, et al. 2013; Vaze et al. 2022).

The open-world supervised learning scenario is not limited to computer vision but applies to numerous domains associated with traditional classification problems (Masana et al. 2022; Rebuffi et al. 2017; Zhu, Ma, et al. 2024). The growing importance of managing novelty in classification tasks, especially in domains like robotics (Toumpa and Cohn 2023) and human activity recognition (Prijatelj et al. 2024), has motivated recent research into open-world learning protocols and benchmarks. It is applicable to any classification problem without a guarantee on the exhaustiveness of the training classes. Indeed, it is plausible that one does not know a priori if a multi-class problem is open or closed. Given this broad spectrum of applications, the open-world condition applies to problems with widely varying data types, sources, and characteristics, including images, text, sensor data, or more structured data types. Indeed, incomplete supervision in these settings can echo the class-ratio problem (Fish and Reyzin 2020), where aggregate label knowledge poses unique modeling challenges.

Emerging real-world applications highlight the critical role of open-world classification in diverse domains. For instance, autonomous vehicles need to identify novel obstacles on the road, medical diagnosis systems must evolve to detect new diseases, and AI chatbots require the ability to process and respond to unknown user intents while minimizing errors (Zhu, Cheng, et al. 2023; Zhu, Ma, et al. 2024). These scenarios underscore the need for models capable of dynamic adaptation, as addressed in open-world classification frameworks; moreover, interpretability concerns in such high-stakes domains further underscore the value of explainable models (Burkart and Huber 2021).

The present work offers a general framework that tackles this issue from a broad perspective. For open-world classification (OWC), ideally, a model is trained on a finite set of known classes and, when applied to new data, it can accurately address four tasks: (i) label all known classes, (ii) identify the instances associated with new, unknown classes, (iii) create new, distinct classes for these instances, and (iv) update itself without losing predictive performance on the known classes while consistently classifying the unknown classes in new data. These tasks are similar to those proposed in earlier work; however, unlike prior approaches that primarily detect unknown instances, this study focuses on automatically categorizing them into distinct classes without manual labeling. Furthermore, to ensure model-agnostic generality, the framework operates on output probability vectors limited to specific machine learning (ML) techniques. Our framework leverages a novel concept of a 'residual signature' (the distinct probability distribution pattern for unknown classes) combined with association rule mining to categorize newly discovered classes automatically.

This study addresses these goals with a framework validated across images, text, sensor data, and structured numeric datasets. The remainder of the paper is organized as follows: Section 2 addresses related work and the state of the art in the field. Section 3 describes the proposed framework, specifically the identification of unknown classes via Jensen-Shannon distance and their categorization using residual signature mining. Section 4 presents comprehensive sensitivity analysis examining the robustness of the framework to key parameters and identifying conditions under which the framework succeeds or fails. Section 5 details experiments conducted across different problem domains with a variety of data and class characteristics. Section 6 demonstrates the

application of the framework through a case study on social media analytics for community resilience. Section 7 discusses the limitations and future work, and Section 8 provides the final concluding remarks.

2 Related Work

In Section 1, the four tasks of open-world classification (OWC) are listed. The first task, creating a classifier to label the known classes, is not the focus of this paper, as it assumes the existence of a pre-trained model capable of handling known classes. The remaining three tasks can be broadly categorized into two groups: identification and categorization. Identification deals with detecting instances associated with unknown classes, while categorization involves organizing these instances into new, distinct classes. Furthermore, the process of updating the model with the capability to classify the new classes is also considered part of the categorization task. The following sections discuss related works addressing these tasks.

2.1 Identification of Unknown Classes

Open set recognition (OSR) constitutes the majority of research related to the identification task. Scheirer, Rezende Rocha, et al. (2013) introduced a mathematical foundation for OSR, modifying a support vector machine to design a 1-vs-set machine. Subsequent work by Scheirer, Jain, et al. (2014) extended this to handle multiple target classes, using a Weibull-Calibrated SVM combined with compact abating probability for unknown class identification. Similarly, adaptations of traditional SVM classifiers for open-world settings were explored by other studies (Jain et al. 2014; Scherreik and Rigling 2016). Moreover, semi-supervised anomaly detection (Görnitz et al. 2013) demonstrates how even partial labels help isolate genuinely novel instances, complementing open-world identification tasks.

In addition to SVM-based approaches, Nearest Class Mean (NCM) classifiers have been proposed for OSR (Bendale and T. Boult 2015; Mensink et al. 2013; Ristin et al. 2014), as well as nearest-neighbor-based approaches (Júnior et al. 2017). These models rely on traditional machine learning techniques but can struggle with complex, high-dimensional data. Interestingly, some anomaly detection algorithms target "zero appearances" of patterns in subspaces (Pang et al. 2016), hinting at how absent patterns might reveal novel classes in open-world scenarios.

More recent efforts leverage deep learning. Bendale and T. E. Boult (2016) replaced the softmax layer in neural networks with an OpenMax layer for OSR. Further advancements include modifying the output layer of deep neural networks (Kardan and Stanley 2017; Shu et al. 2017), learning data representations (Hassen and Chan 2020), and using autoencoders for OSR (Oza and V. M. Patel 2019). Applications span object recognition (Qu et al. 2024), malware detection (Guo et al. 2025), face recognition (Yang et al. 2020), and text classification (Wu et al. 2024). Generative approaches, such as counterfactual image generation with GANs, have also been explored (Ge et al. 2017; Neal et al. 2018).

While this work focuses on classification, recent advances have extended open-world concepts to spatial and temporal tasks. For example, Open World Object Detection (Joseph et al. 2021; Ren et al. 2025) addresses identifying and localizing unknown objects within a scene, while Open-Vocabulary Tracking (R. Li et al. 2025; S. Li et al. 2023) focuses on maintaining object identities over time across unseen categories. Our framework specifically targets the classification stage—operating on probability vectors—which serves as the decision-making core for such downstream tasks

Although these methods offer innovative solutions, most focus on specific domains (often computer vision) and require algorithm-specific modifications, limiting their generalizability.

2.2 Categorization of Unknown Classes

Most OSR research focuses on identifying unknown instances without offering a solution for organizing them into distinct categories. Early efforts, such as (Bendale and T. Boult 2015), relied on manual labeling of identified instances to update models, which is labor-intensive and not scalable.

Automated solutions have begun emerging. Shu et al. (2018) proposed the Pairwise Classification Network (PCN) to determine whether two samples belong to the same class, combining it with hierarchical clustering to categorize observations. Similarly, an unsupervised graph-embedding approach can discover new object affordances in open-set conditions (Toumpa and Cohn 2023), highlighting the potential for domain-agnostic categorization strategies. Alternatively, hierarchical generative models, such as a modified Hierarchical Dirichlet Process (Geng and Chen 2022), estimate the number of clusters in the data. However, these approaches can be sensitive to data distribution assumptions.

Consequently, the lack of accurate, automated methods for categorization remains a key gap in the literature.

2.3 Limitations of Existing Methods

Recent advances in open-world learning have introduced several automated approaches for novel class discovery, yet each has distinct limitations that our framework addresses. The positive–negative prototypes fusion framework by (Zhong and Cui 2025) utilizes contrastive learning to jointly discover unknown categories in open set recognition, but this strategy focuses on visual embeddings and lacks deterministic assignment, limiting its transferability to non-visual or heterogeneous data types. Yan et al. (2025) proposed a CLIP-guided continual novel class discovery approach, leveraging vision-language models for robust adaptation, yet such methods remain tied to deep architectures and large-scale multimodal training data, restricting their use for more general classifier choices and data regimes.

Many existing methods are domain-specific, particularly tailored to computer vision, or they involve algorithmic modifications that limit applicability to diverse data types. Furthermore, categorization methods are either manual or rely on estimations that do not ensure accuracy. Few approaches integrate both identification and categorization of unknown classes in a way that is generally compatible with various data types and classifiers. The proposed framework in this study addresses these limitations by introducing a unified methodology for OWC. It integrates identification and categorization tasks and demonstrates versatility across diverse datasets and classifiers. By providing a generalized approach, it overcomes the narrow applicability of previous methods and offers a scalable solution for open-world classification tasks.

3 Methodology

In this section, we present our framework for open-world classification, which consists of two major stages: (i) identification of unknown classes using anomaly detection on the classifier's probability outputs, and (ii) categorization of flagged instances into newly discovered classes based on residual signatures and association rule mining (ARM). Figure 1 provides a high-level overview of these steps, illustrating how data from known classes are first used to train a probabilistic classifier, which then processes the open-world data. Any observations that do not sufficiently match known classes are flagged as anomalies and passed to the residual signature categorization module, which iteratively creates new class labels and retrains the classifier as needed.

Sections 3.2 and 3.3 elaborate on these components in detail. First, we introduce the notation and the probability space on which our anomaly detection operates. Next, we describe how Jensen-Shannon distance (JSD) identifies unknown instances in O. Finally, we show how the flagged instances are grouped into new classes using frequent itemsets of top probabilities, enabling the framework to adapt dynamically to unknown classes.

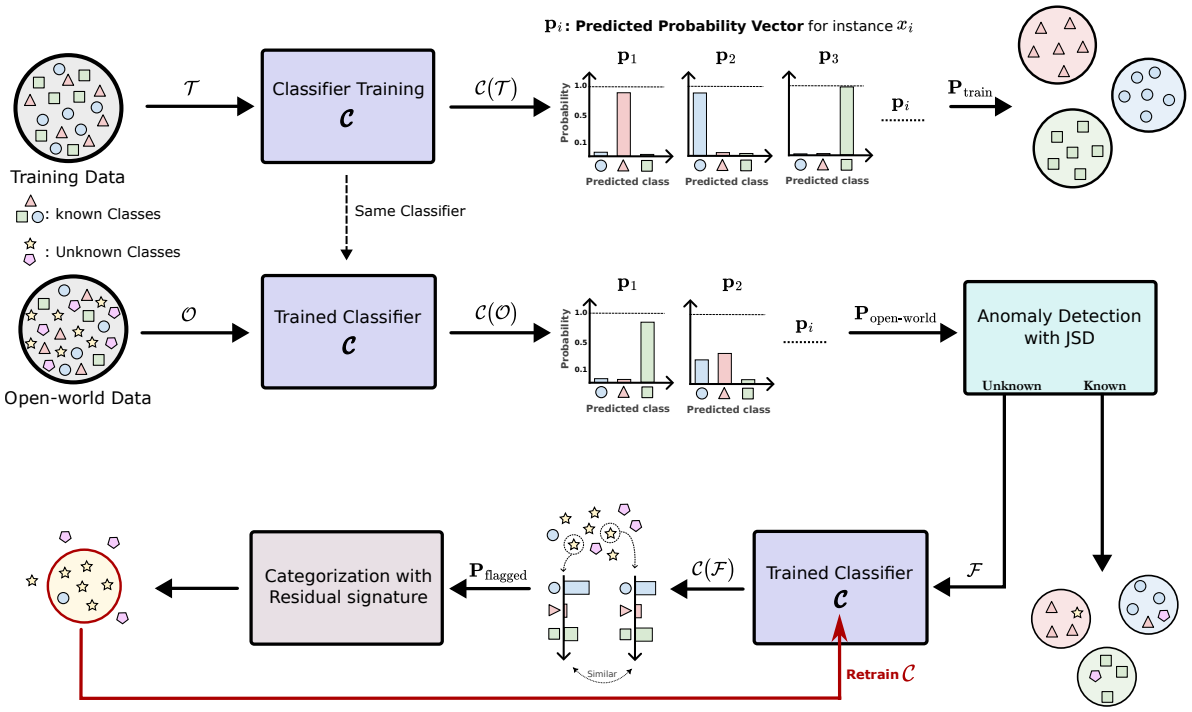

Fig. 1. A high-level overview of the open-world classification framework. A classifier C is first trained on labeled data $\mathcal{T}$ containing known classes (top left). This classifier then processes open-world data O, potentially containing unknown classes, producing a set of predicted probability vectors $\mathbf{p}_i$. An anomaly detection step (via Jensen–Shannon distance) flags any instances that do not map well to known classes ($\mathcal{F}$). These flagged instances are then categorized into new classes by the residual signature categorization module, which uses the most frequent similar probability vectors and association rule mining (ARM) concepts. Newly discovered classes are added to the training set $\mathcal{T}$, and the classifier is retrained, repeating until no further unknown classes are found or stopping criteria are reached.

3.1 Notation

Let $\mathcal{K}$ be the set of known classes and $\mathcal{U}$ the set of unknown classes, where $\mathcal{K} \cap \mathcal{U} = \emptyset$. Let $\mathcal{T}$ denote the training set, which contains n_{T} labeled instances from $k = |\mathcal{K}|$ known classes. Let O be the open-world set, which may contain up to $k + u$ classes, where $u = |\mathcal{U}|$.

Each instance $x_i \in \mathbb{R}^m$ (in either $\mathcal{T}$ or O) is described by m features. The true class of instance i is denoted by $c_i \in \mathcal{K} \cup \mathcal{U}$. A probabilistic classifier C is trained on $\mathcal{T}$ and maps any x_i to a k-dimensional predicted probability vector $\mathbf{p}_i = C(x_i) = (p_{i1}, p_{i2}, \ldots, p_{ik})$, where $p_{ij} \geq 0$ for all $j \in \mathcal{K}$ and $\sum_{j \in \mathcal{K}} p_{ij} = 1$. The predicted class label for instance i is then $\hat{c}_i = \arg\max_{j \in \mathcal{K}} p_{ij}$. We denote by $\mathcal{P}_k$ the k-dimensional output (probability) space to which each $\mathbf{p}_i$ belongs.

3.2 Unknown Class Identification via Jensen–Shannon Distance

The task of identifying unknown classes is necessary when a trained classifier C is applied to O. Many existing works, mostly limited to computer vision, integrate the task of identification with the actual model training, making it a single process. However, to provide a general approach to identify instances in O associated with

classes in $\mathcal{U}$, we separate model training (or algorithm specifications) from the identification task. This is accomplished by performing anomaly detection on the k-dimensional probability space generated by $C(\mathcal{T})$ and $C(O)$. The identification task is now independent of the type of probabilistic classifier and makes no assumptions about the original input data.

The probability distribution generated by $C(\mathcal{T})$ depends on the quality of the classifier and the separability of the classes in the training data. However, the k-dimensional probability space itself is independent of the type of data associated with any problem domain. For a perfect probabilistic classifier, not only is the predicted class for each observation i correct (i.e., $\hat{c}_i = c_i$), but the corresponding probability vector is effectively a one-hot vector (1 for the correct class and 0 for others).

If such a classifier were applied to an observation associated with a class $j \notin \mathcal{K}$, the k-dimensional space would be inadequate to represent the entity. Nonetheless, the observation would still be mapped onto this space, and its projection would likely be a non-binary vector of probabilities. For less-than-perfect classifiers, we operate under the hypothesis that the k-dimensional probability vector for a known class is detectably distinct from that of an unknown class.

There are a variety of metrics to measure dissimilarity between probability vectors, including Kullback–Leibler (KL) divergence (Kullback and Leibler 1951) and Jensen-Shannon distance (JSD) (Lin 1991). For two probability vectors P and Q on the same space X, the KL divergence D_{KL} is computed as

$$D_{\mathrm{KL}}\left(P \| Q\right) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}.$$

The JSD is a symmetric dissimilarity measure derived from the asymmetric KL divergence. For two probability vectors P and Q, it is given by

$$D_{\mathrm{JSD}}\left(P \| Q\right) = \sqrt{\frac{1}{2} D_{\mathrm{KL}}\left(P \| M\right) \; + \; \frac{1}{2} D_{\mathrm{KL}}\left(Q \| M\right)},$$

where $M = \frac{1}{2}(P + Q)$.

The algorithm for identification of unknown classes based on JSD is described in Algorithm 1. First, each instance $x_i \in \mathcal{T}$ is mapped to a probability vector $\mathbf{p}_i \in \mathbb{R}^k$ by the classifier C. Collectively, these probability vectors form an $n \times k$ matrix $\mathbf{P}_{\text{train}}$, where the i-th row corresponds to $\mathbf{p}_i$. For each class $j \in \mathcal{K}$, let $\tilde{c}_j$ be the centroid (mean) of all $\mathbf{p}_i$ whose true class is j. Next, compute the Jensen–Shannon distance $D_{\mathrm{JSD}}(\mathbf{p}_i, \tilde{c}_j)$ for each i, and determine the mean (μ_j) and standard deviation (σ_j) of those distances.

As illustrated in Figure 1, once the classifier C generates probability vectors for the open-world data O, an anomaly detection stage identifies instances that do not map well to any known class. Here we generate the probability vectors for every $x_i \in O$ using C, forming $\mathbf{P}_{\text{open-world}}$. Then, for each $\mathbf{p}_i$ (the i-th row of $\mathbf{P}_{\text{open-world}}$) and each $j \in \mathcal{K}$, compute $D_{\mathrm{JSD}}(\mathbf{p}_i, \tilde{c}_j)$. These values are then mean-centered and scaled by using μ_j and σ_j. For each probability vector $\mathbf{p}_i$, if the scaled $D_{\mathrm{JSD}}(\mathbf{p}_i, \tilde{c}_j)$ is greater than a critical value (d_{critical}) for all $j \in \mathcal{K}$, then i is identified as an anomaly and added to the flagged set $\mathcal{F}$. Otherwise, the instance is considered sufficiently similar to a known class and is not flagged as unknown. The value of d_{critical} is determined using Chebyshev's inequality (Saw et al. 1984) and a confidence parameter α. Since we do not assume any specific underlying distribution for the distances, Chebyshev's inequality provides a robust, distribution-agnostic way to set this threshold.

The flagged set $\mathcal{F}$ generated by Algorithm 1 is used as the primary input to the categorization stage in Algorithm 2, where it is analyzed to determine whether the flagged observations correspond to one or more newly identified unknown classes.

Algorithm 1 Identification of unknown classes (Jensen-Shannon Distance)

Input:
Training data $\mathcal{T}$ with n observations
Open-world data $\mathcal{O}$
Probabilistic classifier C
Confidence level parameter α

Output:
Set $\mathcal{F}$ of flagged observations

$\mathbf{P}_{\text{train}} \leftarrow C(\mathcal{T})$ ▷ An $n_{\mathcal{T}} \times k$ matrix where row i is $\mathbf{p}_i$ for $x_i \in \mathcal{T}$.
for each class $j \in \mathcal{K}$ **do**
  $\tilde{c}_j \leftarrow \text{mean}\left\{\mathbf{p}_i : c_i = j\right\}$ ▷ Centroid of probability vectors for class j.
end for
compute $D_{\text{JSD}}(\mathbf{p}_i, \tilde{c}_j)$ for all i and j, then compute μ_j, σ_j
$\mathbf{P}_{\text{open-world}} \leftarrow C(\mathcal{O})$ ▷ An $n_{\mathcal{O}} \times k$ matrix where row i is $\mathbf{p}_i$ for $x_i \in \mathcal{O}$.
for each $\mathbf{p}_i \in \mathbf{P}_{\text{open-world}}$ **do**
  for each known class $j \in \mathcal{K}$ **do**
    Compute $D_{\text{JSD}}(\mathbf{p}_i, \tilde{c}_j)$, scale using μ_j, σ_j
  end for
  Compute d_{critical} using Chebyshev's inequality with parameter α.
  if scaled $D_{\text{JSD}}(\mathbf{p}_i, \tilde{c}_j) > d_{\text{critical}}$ for all j **then**
    add x_i to $\mathcal{F}$ ▷ x_i is flagged as an anomaly.
  else
    x_i remains unflagged and belongs to a known class.
  end if
end for

3.3 Unknown Class Categorization via Residual Signature Mining

Once observations are identified as not belonging to the set of known classes, the next step is to categorize the instances into new classes. This is the most critical part of the analysis since incorrect categorization of instances can mislead the model while making predictions.

The prediction probabilities associated with the observations are the key drivers for this analysis as they enable generalization of the methodology. When classifier C is applied to observations belonging to classes in $\mathcal{U}$, the resulting k-dimensional output is generally inadequate. The probability vector of an observation belonging to class $j \in \mathcal{K}$ is ideally skewed towards class j in such a way as to exceed some confidence threshold. This is impossible for classes in $\mathcal{U}$. These distributions may either appear noisy or exhibit consistent patterns, since the open-world classifier is incomplete with respect to classes in $\mathcal{U}$. We hypothesize that these patterns are informative and form a distinguishable probability signature for different unknown classes.

We propose a methodology that is in part inspired by association rule mining (ARM) to detect the residual signatures for unknown classes. ARM is a rule-based unsupervised machine learning method to discover interesting relations between variables in large databases (Agrawal et al. 1993; Shahin et al. 2021) and is considered a tool for decision making in marketing (Kaur and Kang 2016), medical diagnosis (P. Patel et al. 2024), bioinformatics (Shi et al. 2024), and other fields. While alternative techniques such as K-means, K-medoids, or K-NN matching

could be used in place of ARM, we argue that ARM offers distinct advantages in this context. The probability vectors associated with unknown classes contain substantial noise, which can lead to instability in clustering and other traditional pattern-matching approaches. The proposed methodology mitigates this issue by discretizing these vectors and treating them as itemsets. This enables evaluation and categorization to be driven by the most relevant classifier probabilities, rather than by diffuse and noisy uncertainty in predictions for unseen classes.

ARM accomplishes two distinct tasks: (i) identification of frequent itemsets within the data and (ii) generation of rules associated with the frequent itemsets. The concept of frequent itemset analysis is integrated into the algorithm to discover the unknown classes. Figure 1 depicts this process, where flagged instances in $\mathcal{F}$ from Algorithm 1 are grouped into new classes based on their residual signatures. The residual signature is a k-dimensional probabilistic vector generated by the classifier C. This residual signature is transformed into itemsets by selecting the class labels corresponding to the top-τ probabilities of the vector. We propose these itemsets (the top-τ probabilities in each $\mathbf{p}_i$) as the distinguishing itemset for a given unknown class. Observations belonging to the same unknown class should exhibit similar top-τ probabilities.

The algorithm for categorization of identified observations is described in Algorithm 2. The required inputs include the sets $\mathcal{T}$, $\mathcal{F}$, O, and the classifier C. The parameter τ defines the itemset size (i.e., classes that form itemsets) to characterize the flagged instances. The parameter $0 \leq \gamma \leq 1$ is a prediction probability threshold. The parameters ω, η, and β take values between 0 and 1 and govern the stopping criteria. Specifically, η imposes a minimum size for a newly discovered class (if $|\mathcal{B}| \leq \eta|\mathcal{F}|$, the algorithm terminates to avoid labeling a too-small cluster). The parameter ω controls the fraction of flagged data that justifies further attempts at discovering additional unknown classes, while β dictates when to stop pulling in remaining instances of a newly discovered class.

The algorithm is iterative, and the unknown classes are discovered one at a time sequentially. For every unknown class discovered, the instances associated with that particular class are identified and removed from the flagged set $\mathcal{F}$. This process continues until there are fewer than ω percent of observations remaining in $\mathcal{F}$.

The first step is to find the prediction probabilities of all the flagged observations. Let $\mathbf{P}_{\text{flagged}}$ be the $m \times k$ probability matrix obtained by applying C to each flagged instance $x_i \in \mathcal{F}$. That is, row i of $\mathbf{P}_{\text{flagged}}$ is $\mathbf{p}_i$. For each $\mathbf{p}_i$, select the top-τ probabilities to form an itemset. We then apply association rule mining (ARM) to find the frequent itemsets of size τ. If the most frequent itemset has sufficient support, the corresponding instances are labeled as a new class $k + i$ (i is incremented for every unknown class discovered). Table 1 illustrates an example scenario with 5 flagged observations and 3 known classes and highlights how top-τ probabilities form an itemset. In this table, index 1 is associated with the first flagged observation for which classes c_1 and c_2 have the highest probabilities across the residual signature (assuming $\tau = 2$). Similarly, the highest τ probabilities for each observation in $\mathbf{P}_{\text{flagged}}$ are determined and the corresponding training class labels are stored as items in a set $\mathcal{I}$.

Determine the itemset s with the greatest support (frequency) and identify all the instances associated with s; add them to the set $\mathcal{B}$. If the number of elements in $\mathcal{B}$ is less than $\eta|\mathcal{F}|$, the algorithm terminates without updating C since there were insufficient observations to confidently retrain the model. The newly labeled observations are removed from $\mathcal{F}$ and added to the training set $\mathcal{T}$. Retrain the classifier C on $\mathcal{T}$.

The set $\mathcal{B}$ may not contain all instances of class $k + i$ from $\mathcal{F}$. Therefore, the classifier is repeatedly applied to $\mathcal{F}$ in order to attempt to capture the remaining instances. During each iteration, as new observations are associated with class $k + i$ (with a prediction probability threshold of at least γ), they are removed from the flagged set, added to the training set, and the classifier is updated. This continues until there are only β percent of flagged instances predicted as $k + i$. If $\beta = 0$, then there are no instances in $\mathcal{F}$ predicted as class $k + i$.

The process repeats creating a new itemset $\mathcal{I}$ based on the updated classifier C applied to the updated flagged set $\mathcal{F}$. The algorithm terminates when there are only ω percent of the original number of flagged observations remaining to be classified. The final output is a classifier capable of classifying both the original known classes and all of the identified unknown classes.

Algorithm 2 Residual signature categorization

Input:
Training data $\mathcal{T}$ with n observations
Set $\mathcal{F}$ of m flagged observations
Open-world data $\mathcal{O}$
Probabilistic classifier C
τ = itemset size
Prediction probability parameter γ
Termination parameters ω, η, and β

Output:
Updated probabilistic classifier C

$L \leftarrow |\mathcal{F}|$
$i \leftarrow 0$
while $|\mathcal{F}| > \omega L$ **do**
  $\mathbf{P}_{\text{flagged}} \leftarrow C(\mathcal{F})$ ▷ An $m \times k$ matrix where row i is $\mathbf{p}_i$ for $x_i \in \mathcal{F}$.
  create $\mathcal{I}$ from $\mathbf{P}_{\text{flagged}}$ based on τ
  $\mathcal{B} \leftarrow$ instances in $\mathcal{F}$ associated with the itemset having the highest support
  if $|\mathcal{B}| \leq \eta|\mathcal{F}|$ **then return** C
  end if
  $i \leftarrow i + 1$
  create new label $k + i$ for all observations in $\mathcal{B}$
  $\mathcal{F} \leftarrow \mathcal{F} \setminus \mathcal{B}$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{B}$
  retrain the classifier C on $\mathcal{T}$
  repeat
    $\mathcal{P} \leftarrow$ instances in $\mathcal{F}$ predicted as $k + i$ with probability at least γ
    $\mathcal{F} \leftarrow \mathcal{F} \setminus \mathcal{P}$
    $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{P}$
    retrain the classifier C on $\mathcal{T}$
  until $|\mathcal{P}| \leq \beta\,|\mathcal{F}|$
end while
return C

4 Sensitivity Analysis

The proposed framework involves several parameters that govern its behavior during the identification and categorization stages. To evaluate the robustness of the methodology and provide practical guidance for parameter selection, we conduct systematic sensitivity analysis on four key factors that may influence performance. Specifically, we examine: (i) the number of known classes in the training set, (ii) the critical distance threshold determined by Chebyshev's inequality, (iii) the itemset size parameter that defines residual signature length, and (iv) the quality of the base classifier. Building on these findings, we provide a set of parameter selection guidelines to assist practitioners in deploying the framework. To maintain consistency and enable direct comparison across experiments, all sensitivity analyses are conducted using the MNIST handwritten digit dataset with a Random Forest classifier as the base model.

Table 1. This table shows 5 flagged observations, their predicted probabilities for classes c_1, c_2, c_3, the resulting 2-item sets, and the final assigned new class (c_4).

Flagged observations in $\mathcal{F}$	$\mathbf{P}_{\text{flagged}}$			Itemset in $\mathcal{I}$	Labeling the most frequent Itemsets
	Class c_1	Class c_2	Class c_3		
x_1	**0.40**	**0.50**	0.10	$\{c_1, c_2\}$	
x_2	**0.70**	0.10	**0.20**	$\{c_1, c_3\}$	label as a new class c_4
x_3	**0.61**	0.09	**0.30**	$\{c_1, c_3\}$	label as a new class c_4
x_4	0.05	**0.75**	**0.20**	$\{c_2, c_3\}$	
x_5	**0.65**	0.03	**0.32**	$\{c_1, c_3\}$	label as a new class c_4

For each sensitivity analysis, we systematically vary a single parameter while holding all others constant, enabling isolation of individual parameter effects. Unless otherwise specified, we use the following default parameter settings: Chebyshev confidence parameter $\alpha = 0.1$ yielding critical distance $d_{\text{critical}} = 3.16$, termination parameters $\omega = 0.25$, $\eta = 0.1$, $\beta = 0.1$, and prediction confidence threshold $\gamma = 0.6$. The open-world dataset contains instances from all digits 0 through 9, with specific digits designated as unknown classes for each experiment. Performance is evaluated using accuracy, precision, recall, and F_1-score metrics, with particular attention to the accuracy achieved on each unknown class and the total number of unknown classes successfully discovered.

4.1 Impact of Number of Known Classes

The dimensionality of the probability space is determined by the number of known classes $k = |K|$. A richer probability space may provide more detailed information for distinguishing between different unknown classes based on their residual signatures. We hypothesize that as the number of known classes increases, the framework's ability to categorize unknown classes improves due to the availability of more distinguishing features in the probability vectors.

4.1.1 Experimental Setup. We conduct three experiments varying $|K| \in \{2, 5, 8\}$ while maintaining $|U| = 2$ constant across all experiments. Specifically, digits 8 and 9 are designated as unknown classes in all three experiments. For Experiment 1, the training set contains only digits 0 and 1 ($|K| = 2$). For Experiment 2, the training set includes digits 0 through 4 ($|K| = 5$). For Experiment 3, the training set comprises digits 0 through 7 ($|K| = 8$). The parameter τ is adjusted proportionally to the number of known classes following the guideline $\tau \approx |K|/2$, yielding $\tau = 1$, 3, and 4 for the three experiments respectively. The prediction confidence parameter γ is set to 0.6 for Experiments 1 and 2, and reduced to 0.4 for Experiment 3 to account for the increased complexity. Table 2 summarizes the experimental configuration.

4.1.2 Results. The results across the three experiments reveal a clear trend: framework performance improves substantially as the number of known classes increases. Table 3 presents the aggregated results for all three experiments, organized by the identification stage, categorization stage, and overall performance metrics.

In Experiment 1 with only 2 known classes, approximately 50% of open-world instances (2,041 out of 4,000) were flagged by Algorithm 1, with 96% of flagged instances belonging to the two unknown classes, indicating strong identification capability at this stage. However, Algorithm 2 discovered only a single unknown class rather than two, revealing a critical failure in categorization. Examining the detailed results shows that 989 out of 1,005 instances of digit 8 (98%) were correctly classified as the discovered unknown class. In contrast, all 995 instances of digit 9 were misclassified across various classes, with the majority (977 instances) incorrectly assigned to the

Table 2. Experimental Configuration for Known Class Sensitivity Analysis

Parameter	Experiments		
	1	2	3
Number of unknown classes ($\|U\|$)	2	2	2
Number of known classes ($\|K\|$)	2	5	8
Unknown classes (digits)	8, 9	8, 9	8, 9
Known classes (digits)	0, 1	0–4	0–7
Classifier	RF	RF	RF
Chebyshev confidence (α)	0.1	0.1	0.1
Critical distance (d_{critical})	3.16	3.16	3.16
Itemset size parameter (τ)	1	3	4
Termination parameter (ω)	0.25	0.25	0.25
Termination parameter (β)	0.1	0.1	0.1
Termination parameter (η)	0.1	0.1	0.1
Prediction confidence (γ)	0.6	0.6	0.4

Table 3. Performance Comparison Across Different Numbers of Known Classes

Metric	$\|K\|$		
	2	5	8
Identification Stage (Algorithm 1)			
Percentage of O flagged	50	46	50
Purity of $\mathcal{F}$ (% from unknown)	96	88	84
Categorization Stage (Algorithm 2)			
Unknown classes discovered	1†	2	2
Unknown Class Performance			
Class 8 correct (count/total)	989/1005	641/1005	743/1005
Class 8 accuracy (%)	98	64	74
Class 9 correct (count/total)	0/995	731/995	787/995
Class 9 accuracy (%)	0†	73	79
Overall Performance			
Final accuracy	0.73	0.81	0.86
Final F_1-score	0.65	0.81	0.87

† Experiment 1: Framework failed to separate the 2 unknown classes

single discovered unknown class along with digit 8. This failure to distinguish between the two unknown classes resulted in an overall F_1-score of 0.65.

In Experiment 2 with 5 known classes, approximately 46% of open-world instances were flagged, with 88% purity (belonging to unknown classes). Algorithm 2 successfully discovered both unknown classes as distinct entities. Classification accuracy for digit 8 was 64% (641 out of 1,005 correct), with most misclassifications

distributed between the newly discovered class for digit 9 and known classes. For digit 9, accuracy reached 73% (731 out of 995 correct), with most misclassifications assigned to known class 4. The overall F_1-score improved substantially to 0.81, representing a 16-point gain over Experiment 1.

In Experiment 3 with 8 known classes, approximately 50% of instances were flagged, with 84% purity. Both unknown classes were successfully discovered and separated. Classification accuracy for digit 8 increased to 74% (743 out of 1,005 correct), with major misclassifications to known class 3 due to visual similarity in curved features. For digit 9, accuracy reached 79% (787 out of 995 correct), with most errors assigned to known class 4. The overall F_1-score improved further to 0.87, representing a 6-point gain over Experiment 2 and a 22-point gain over Experiment 1.

4.1.3 Discussion. The results strongly support the hypothesis that increasing the number of known classes improves the framework's categorization performance. With only 2 known classes, the 2-dimensional probability space provides insufficient information for Algorithm 2 to distinguish between the two unknown classes. Both digits 8 and 9 exhibited similar probability patterns when projected onto this limited space, causing the frequent itemset mining to identify a single dominant pattern shared by both unknown classes. This explains why the identification stage succeeded (96% purity in flagged instances) but the categorization stage failed (discovering only 1 instead of 2 classes).

As the dimensionality increases to 5 and then 8, the probability vectors for digits 8 and 9 exhibit increasingly distinct patterns. With 5 known classes, digit 8's probability vector shows higher weights on classes that share curved features (such as digit 3), while digit 9's vector shows affinity toward classes with vertical structures (such as digit 4). These distinct residual signatures enable successful separation into two discovered classes.

The improvement from $|K| = 5$ to $|K| = 8$ is more modest (F_1-score from 0.81 to 0.87, a 6-point gain) compared to the dramatic improvement from $|K| = 2$ to $|K| = 5$ (F_1-score from 0.65 to 0.81, a 16-point gain). This suggests diminishing returns beyond a certain threshold, although classification accuracy on individual unknown classes continues to improve with additional known classes (digit 8: 64% $\rightarrow$ 74%; digit 9: 73% $\rightarrow$ 79%).

An important practical implication emerges: the framework performs best when $|K| \geq |U|$. When the number of unknown classes approaches or exceeds the number of known classes (as in Experiment 1 where $|U|/|K| = 1$), performance degrades substantially due to insufficient dimensionality in the probability space. For applications where the ratio $|U|/|K|$ is expected to be high, practitioners should consider whether the framework is appropriate, or whether collecting training data for additional known classes is feasible before deployment.

4.2 Impact of Chebyshev Inequality Parameter

The Chebyshev inequality parameter determines the critical distance threshold for identifying anomalies in Algorithm 1. This threshold is derived from the confidence parameter α using Chebyshev's inequality, which provides distribution-free bounds. We hypothesize that the framework should exhibit robustness to reasonable variations in d_{critical} because: (i) Chebyshev's inequality provides conservative guarantees regardless of the underlying distribution, and (ii) Algorithm 2's categorization stage can filter false positives flagged by Algorithm 1. However, we also expect that extreme values may cause failure—very low values may flag excessive false positives, while very high values may miss true unknown instances.

4.2.1 Experimental Setup. We conduct three experiments varying $d_{\text{critical}} \in \{1.414, 2.0, 4.47\}$, which correspond to confidence levels $\alpha \approx \{0.5, 0.25, 0.05\}$ respectively. The training set contains 8 known classes (digits 0–7) and the open-world data contains 2 unknown classes (digits 8 and 9), consistent with Experiment 3 from the previous analysis. All other parameters remain constant: $\tau = 4$, $\omega = 0.25$, $\eta = 0.1$, $\beta = 0.1$, $\gamma = 0.4$. Table 4 summarizes the configuration.

Table 4. Experimental Configuration for Chebyshev Parameter Sensitivity Analysis

Parameter	Experiments		
	1	2	3
Known classes (digits)	0–7	0–7	0–7
Unknown classes (digits)	8, 9	8, 9	8, 9
Classifier	RF	RF	RF
Critical distance (d_{critical})	1.414	2	4.47
Confidence level (α)	0.50	0.25	0.05
Itemset size parameter (τ)	4	4	4
Termination parameter (ω)	0.25	0.25	0.25
Termination parameter (β)	0.1	0.1	0.1
Termination parameter (η)	0.1	0.1	0.1
Prediction confidence (γ)	0.4	0.4	0.4

4.2.2 Results. The results demonstrate remarkable robustness to variations in d_{critical}, with final classification performance remaining nearly identical across all three experiments despite substantial differences in the identification stage. Table 5 presents the comprehensive results, organized by identification stage outcomes and final performance metrics.

Table 5. Performance Comparison Across Different Chebyshev Parameters

Parameter	Experiments		
	1	2	3
Critical distance (d_{critical})	1.414	2	4.47
Confidence level (α)	50%	25%	5%
Identification Stage (Algorithm 1)			
Total instances flagged ($\lvert\mathcal{F}\rvert$)	2,615	2,502	2,352
Percentage of O flagged	65	62	59
Flagged from known classes	619	500	388
Flagged from unknown classes	1,996	2,002	1,964
Purity (% from unknown)	76	80	84
Categorization & Performance			
Unknown classes discovered	2	2	2
Class 8 correct (count/total)	530/1005	530/1005	530/1005
Class 8 accuracy (%)	53	53	53
Class 9 correct (count/total)	838/995	838/995	821/995
Class 9 accuracy (%)	84	84	83
Final F_1-score	**0.81**	**0.81**	**0.81**

As expected, lower critical distance values result in more instances being flagged. With $d_{\text{critical}} = 1.414$, approximately 65% of the open-world data (2,615 out of 4,000 instances) was flagged, compared to 59% (2,352

instances) with $d_{\text{critical}} = 4.47$. The additional flagged instances with lower thresholds are predominantly false positives from known classes, as evidenced by the purity of the flagged set decreasing from 84% (with $d_{\text{critical}} = 4.47$) to 76% (with $d_{\text{critical}} = 1.414$). In absolute terms, Experiment 1 flagged 619 instances from known classes compared to only 388 in Experiment 3, representing a 59% increase in false positives.

However, this substantial increase in false positives does not degrade final performance. In all three experiments, Algorithm 2 successfully discovered both unknown classes, and the classification accuracy on each unknown class remained nearly identical: digit 8 achieved exactly 53% accuracy (530 out of 1,005 correct) in all three experiments, while digit 9 achieved 84–85% accuracy (838–841 out of 995 correct). The overall F_1-score remained constant at 0.81 for all three parameter settings, demonstrating complete robustness to this parameter choice within the tested range.

4.2.3 Discussion. The remarkable stability of performance across widely varying d_{critical} values (spanning a 3.2-fold range from 1.414 to 4.47) demonstrates that the framework is robust to the choice of Chebyshev parameter. This robustness can be attributed to two factors.

First, while Algorithm 1 flags more instances with lower threshold values, the additional flagged instances are primarily false positives that do not share the residual signatures of true unknown classes. These false positives fail to participate in the frequent itemsets identified by Algorithm 2 because their probability patterns differ from both unknown classes. For example, a known-class instance incorrectly flagged as unknown might have high probability for its true class and low probabilities elsewhere, creating an itemset pattern distinct from the diffuse patterns characteristic of unknown classes. This causes false positives to effectively self-filter from the categorization process.

Second, the iterative refinement process in Algorithm 2 (lines 14–19) progressively removes correctly classified instances from $\mathcal{F}$, whether they are true unknowns or false positives. During each iteration, the retrained classifier correctly predicts many false positives as their true known classes (with high confidence γ), removing them from $\mathcal{F}$ before the next unknown class discovery cycle. This refinement converges to similar final states regardless of the initial composition of $\mathcal{F}$.

The nearly perfect consistency in discovering exactly 2 unknown classes across all experiments further validates the categorization approach. Even with 76% purity in Experiment 1 (meaning 619 false positives among 2,615 flagged instances, representing nearly one-quarter noise), Algorithm 2 correctly identified two distinct unknown classes without creating spurious additional classes from the false positive noise.

From a practical standpoint, this robustness is highly desirable. Practitioners do not need to carefully tune d_{critical} or conduct extensive validation to select α. Based on these results, we recommend using the default setting $\alpha = 0.1$ (yielding $d_{\text{critical}} \approx 3.16$), which provides a reasonable balance between sensitivity and specificity. This setting allows up to 10% of known-class instances to potentially be flagged while maintaining high purity (84% in our experiments), and the framework demonstrates that it can handle even lower purity (76%) without performance degradation.

An important caveat: while the framework is robust to d_{critical} within the tested range [1.414, 4.47], extreme values outside this range have not been evaluated. Extremely low d_{critical} (approaching 1.0) might flag the majority of known-class instances, potentially overwhelming Algorithm 2 with excessive noise and causing computational issues. Extremely high d_{critical} (beyond 5.0) might fail to flag sufficient unknown instances, particularly for unknown classes with subtle residual signatures that produce probability patterns only marginally different from known classes. We recommend keeping d_{critical} in the range [1.5, 4.5] for practical applications.

4.3 Impact of Itemset Size Parameter

The itemset size parameter τ determines the number of top probability classes used in Algorithm 2 for categorizing unknown classes. Each flagged instance generates an itemset consisting of the class labels corresponding to

its top-τ probability values. These itemsets represent the residual signature patterns that distinguish different unknown classes.

We hypothesize that extreme values of τ will cause framework failure, while moderate values will succeed. Specifically: (i) very low τ provides insufficient information to capture the full residual signature, causing multiple unknown classes to appear similar and preventing proper separation; (ii) very high τ incorporates excessive information including noise from low-probability classes, creating overly general patterns shared across multiple unknown classes; (iii) moderate τ values (approximately $|K|/2$) provide optimal trade-off between informativeness and specificity.

4.3.1 Experimental Setup. We conduct three experiments varying $\tau \in \{1, 4, 7\}$ while maintaining all other parameters constant. The training set contains 8 known classes (digits 0–7) and the open-world data contains 2 unknown classes (digits 8 and 9), consistent with previous analyses. To isolate the effect of τ, we first apply Algorithm 1 once to generate the flagged set $\mathcal{F}$ (with $|\mathcal{F}| = 3{,}543$ instances, 83% purity), then use this same $\mathcal{F}$ across all three experiments, varying only the τ parameter in Algorithm 2. This ensures that differences in performance are attributable solely to τ rather than variations in the identification stage. Table 6 summarizes the configuration.

Table 6. Experimental Configuration for Itemset Size Parameter Sensitivity Analysis

Parameter	Experiments		
	1	2	3
Known classes (digits)	0–7	0–7	0–7
Unknown classes (digits)	8, 9	8, 9	8, 9
Classifier	RF	RF	RF
Chebyshev confidence (α)	0.1	0.1	0.1
Critical distance (d_{critical})	3.16	3.16	3.16
Itemset size parameter (τ)	**1**	**4**	**7**
Shared flagged set ($\|\mathcal{F}\|$)	3,543	3,543	3,543
Purity of $\mathcal{F}$	83%	83%	83%
Termination parameters	$\omega = 0.25$, $\eta = 0.1$, $\beta = 0.1$, $\gamma = 0.4$		

4.3.2 Results. The results confirm the hypothesis that τ significantly impacts categorization performance, with extreme values causing substantial degradation. Table 7 presents the comprehensive results, showing performance on both known classes (to assess degradation) and unknown classes (to assess discovery quality).

With $\tau = 1$, itemsets consist only of the single highest probability class for each flagged instance. Algorithm 2 discovered three unknown classes rather than the correct two, indicating failure to properly categorize the unknowns. Examining the detailed results reveals that digit 8 was split across multiple discovered classes: 552 instances (37%) were assigned to one discovered class, 581 instances (39%) were misclassified as the discovered class for digit 9, and 201 instances (13%) were assigned to a spurious third class. The performance on digit 9 was relatively good at 93% accuracy (1,400 out of 1,510 correct), as most digit 9 instances shared the same top-probability class, making them easier to group. However, the overall F_1-score of 0.77 was substantially lower than optimal.

With $\tau = 4$, Algorithm 2 correctly discovered exactly two unknown classes. Classification accuracy for digit 8 improved dramatically to 66% (986 out of 1,490 correct), with most remaining misclassifications to digit 3 (134 instances) due to visual similarity in curved features. Only 116 instances (8%) of digit 8 were misclassified as digit

Table 7. Performance Comparison Across Different Itemset Size Parameters

Metric	τ		
	1	4	7
Discovery Results			
Unknown classes discovered	3†	2	2
Known Class Performance			
Class 0 accuracy	99% (365/368)	99% (365/368)	91% (333/368)
Class 1 accuracy	99% (441/445)	93% (412/445)	90% (399/445)
Unknown Class Performance			
Class 8 correct (count/total)	552/1490	986/1490	202/1490
Class 8 accuracy (%)	37	**66**	13
Class 8 misclassified as class 9	581 (39%)	116 (8%)	1154 (77%)
Class 9 correct (count/total)	1400/1510	1306/1510	1357/1510
Class 9 accuracy (%)	93	86	90
Overall Performance			
Final F_1-score	0.77	**0.86**	0.69

† Experiment 1: Framework incorrectly split unknown classes into 3 groups

9, representing an 80% reduction compared to $\tau = 1$. Digit 9 achieved 86% accuracy (1,306 out of 1,510 correct), with primary misclassifications to digit 4 (93 instances). Known-class performance remained high with classes 0 and 1 achieving 99% and 93% accuracy respectively. This experiment achieved the highest overall F_1-score of 0.86, demonstrating optimal performance.

With $\tau = 7$, itemsets contain nearly all 8 known classes (7 out of 8), creating overly general patterns. While Algorithm 2 discovered two classes (correct count), the categorization quality was poor. Only 13% of digit 8 instances (202 out of 1,490) were correctly classified, with the vast majority—1,154 instances (77%)—misclassified as digit 9, representing a nearly 10-fold increase compared to $\tau = 4$. This occurred because the 7-element itemsets for both digit 8 and digit 9 contained similar sets of classes, causing Algorithm 2 to group them together. Digit 9 achieved 90% accuracy (1,357 out of 1,510), benefiting from absorbing many digit 8 instances into its discovered class. However, the overall F_1-score dropped to 0.69, representing a 20% degradation from optimal. Additionally, known-class accuracy degraded substantially, with class 0 dropping to 91% (from 99%) and class 1 to 90% (from 93%), indicating that the excessive noise in 7-element itemsets also confused the retrained classifier on previously well-learned classes.

4.3.3 Discussion. The results clearly demonstrate that τ is a critical parameter requiring careful selection. The failure modes at extreme values align with theoretical expectations and provide insights into the mechanics of residual signature mining.

With $\tau = 1$, each instance's residual signature is characterized by only its highest probability class. This single feature is insufficient to distinguish between unknown classes that may share similarities with the same known class. For example, both digits 8 and 9 might assign highest probability to digit 3 due to curved features, making them appear identical in the itemset space. The discovery of 3 classes instead of 2 suggests that some subset of digit 8 instances had different highest-probability classes (perhaps some favored digit 3 while others favored digit

0), causing Algorithm 2 to split them incorrectly into separate groups. This explains the 37% accuracy on digit 8 and the creation of a spurious third class.

With $\tau = 7$ (approaching $|K| = 8$), itemsets become overly inclusive. When an itemset contains 7 out of 8 classes, it captures not just the distinguishing signature but also irrelevant noise from classes with low probabilities (e.g., 2–5% probabilities). Furthermore, the complement principle applies: specifying 7 classes is nearly equivalent to specifying which single class is NOT in the top 7, losing the rich pattern information available in the probability distribution's shape. Different unknown classes that might have distinct patterns in their top 3–4 probabilities become indistinguishable when considering their top 7 probabilities, as these long itemsets converge to similar sets containing all major and minor probability masses.

The optimal performance at $\tau = 4 = |K|/2$ suggests that approximately half the known classes provide sufficient information to capture discriminative patterns while avoiding noise. This finding aligns with information-theoretic principles: τ should be large enough to capture the shape of the probability distribution (requiring multiple points) but small enough to focus on the most informative components (excluding the noisy tail).

For practical guidance, we recommend setting $\tau \approx \lfloor |K|/2 \rfloor$, with acceptable range $[\lfloor |K|/3 \rfloor, \lfloor 2|K|/3 \rfloor]$. Practitioners should avoid $\tau < 2$ (insufficient information) and $\tau > 3|K|/4$ (excessive noise). If unsure, erring toward slightly lower τ is preferable to excessively high τ, as the results show that $\tau = 1$ ($F_1 = 0.77$) outperformed $\tau = 7$ ($F_1 = 0.69$) despite both being suboptimal, likely because lower values create cleaner (albeit incomplete) signatures while higher values create overly polluted signatures.

4.4 Impact of Base Classifier Quality

The framework operates on probability vectors generated by a base classifier, relying on the hypothesis that probability patterns for known classes are detectably distinct from patterns for unknown classes. This hypothesis implicitly assumes a certain level of base classifier quality.

We hypothesize that classifier quality critically affects framework success through the following mechanism: high-quality classifiers produce confident, accurate predictions on known classes (probability vectors close to one-hot encoding) but uncertain, diffuse predictions on unknown classes (probability vectors spread across multiple classes). This difference creates detectable anomalies in Algorithm 1. In contrast, low-quality classifiers produce uncertain predictions on both known and unknown classes, eliminating the distinguishable difference that the framework relies upon. Therefore, we expect the framework to require a minimum threshold of base classifier quality to function effectively.

4.4.1 Experimental Setup. We conduct three experiments with base classifiers of varying quality: high (99% accuracy), moderate (70% accuracy), and low (51% accuracy). To control classifier quality, we train Random Forest classifiers with different hyperparameters and training set sizes. The open-world data remains constant across experiments, containing instances with 8 known classes (digits 0–7) and 2 unknown classes (digits 8 and 9). All framework parameters are held constant: $\alpha = 0.1$ yielding $d_{\text{critical}} = 3.16$, $\tau = 4$, $\omega = 0.25$, $\eta = 0.1$, $\beta = 0.1$, $\gamma = 0.4$. Table 8 summarizes the configuration.

4.4.2 Results. The results dramatically confirm the hypothesis that base classifier quality is critical for framework success. Table 9 presents the stark differences in performance.

With the high-quality classifier (99% accuracy), the framework functions as expected. Algorithm 1 successfully flags 49% of open-world instances (1,964 out of 4,000), with high purity indicating most flagged instances belong to unknown classes. Algorithm 2 discovers both unknown classes, achieving overall F_1-score of 0.86, consistent with previous experiments in Sections 4.2 and 4.3.

With moderate-quality (70%) and low-quality (51%) classifiers, the framework experiences complete failure at the first stage. Algorithm 1 flags zero instances as anomalies. Without any flagged instances, Algorithm 2 cannot

Table 8. Experimental Configuration for Base Classifier Quality Sensitivity Analysis

Parameter	Experiments		
	1	2	3
Known classes (digits)	0–7	0–7	0–7
Unknown classes (digits)	8, 9	8, 9	8, 9
Classifier	RF	RF	RF
Base accuracy on known classes	**High (99%)**	**Moderate (70%)**	**Low (51%)**
Chebyshev confidence (α)	0.1	0.1	0.1
Critical distance (d_{critical})	3.16	3.16	3.16
Itemset size parameter (τ)	4	4	4
Termination parameters	$\omega = 0.25$, $\eta = 0.1$, $\beta = 0.1$, $\gamma = 0.4$		

Table 9. Performance Comparison Across Different Base Classifier Quality Levels

Parameter	Experiments		
	1	2	3
Base accuracy on known classes	High (99%)	Moderate (70%)	Low (51%)
Identification Stage (Algorithm 1)			
Instances flagged ($\lvert\mathcal{F}\rvert$)	1,964	0	0
Percentage of O flagged	49	0†	0†
Algorithm 1 outcome	Success	**Failure**	**Failure**
Categorization & Performance			
Unknown classes discovered	2	–	–
Final F_1-score	0.86	–	–
Framework outcome	**Success**	**Complete Failure**	**Complete Failure**

† Framework terminated immediately; no instances flagged as anomalies

proceed, and no unknown classes are discovered. The framework terminates immediately without updating the classifier or discovering any of the unknown classes present in the open-world data. This represents a total breakdown of the identification mechanism.

4.4.3 Discussion. The complete failure of the framework with moderate and low-quality classifiers reveals a fundamental limitation: the framework requires high-quality base classifiers to function. This requirement stems from the core mechanism of Algorithm 1, which detects anomalies by comparing probability patterns to centroids of known classes. This comparison is meaningful only when: (i) known-class instances produce consistent, confident probability patterns centered on the correct class, (ii) unknown-class instances produce detectably different (more diffuse, less confident) patterns, and (iii) the difference between these two types of patterns exceeds natural variation within known classes.

When classifier quality degrades below approximately 70% accuracy, condition (iii) fails. The natural variation in probability patterns for known-class instances (due to classifier errors and uncertainty) becomes comparable to or exceeds the difference between known and unknown patterns. To illustrate, a high-quality classifier (99% accuracy)

produces probability vectors like $[0.01, 0.02, \mathbf{0.94}, 0.01, 0.01, 0.00, 0.01, 0.00]$ for known-class instances (near-one-hot) and $[0.03, 0.05, 0.14, \mathbf{0.31}, 0.06, 0.11, 0.04, 0.26]$ for unknown-class instances (diffuse). The Jensen-Shannon distance between these patterns and the class centroids differs dramatically. In contrast, a moderate-quality classifier (70% accuracy) produces patterns like $[0.08, 0.11, \mathbf{0.43}, 0.09, 0.12, 0.04, 0.09, 0.04]$ for known classes and $[0.06, 0.09, 0.17, \mathbf{0.24}, 0.11, 0.13, 0.07, 0.13]$ for unknown classes—both are similarly diffuse, making discrimination impossible.

In effect, the "signal" (difference between known and unknown patterns) is drowned out by the "noise" (variation within known-class patterns). This explains why zero instances are flagged: all instances, whether known or unknown, exhibit similarly high distances from class centroids when the base classifier is of poor quality.

This finding has critical practical implications. Before applying this framework to a new problem domain, practitioners must first train a high-quality classifier on the known classes. If the base classifier cannot achieve at least 70% accuracy (preferably exceeding 90%), the framework should not be applied. This may indicate that: (i) the feature representation is inadequate, (ii) the known classes are not sufficiently separable, (iii) more training data is needed, or (iv) a different classifier type should be considered.

The framework is best suited for domains where high-quality classification is achievable but open-world scenarios are common. Examples include image classification with clear visual distinctions (MNIST, CIFAR-10), text classification with well-defined topics, and sensor data with distinct activity patterns. The framework is poorly suited for domains with inherently noisy, ambiguous data where even state-of-art classifiers struggle to exceed 70% accuracy, such as fine-grained emotion recognition from text or early disease detection from limited symptoms.

4.5 Parameter Selection Guidelines

Based on the sensitivity analysis findings, we provide practical recommendations for parameter selection to guide practitioners applying the framework to new domains.

Chebyshev Confidence Parameter (α): Set $\alpha = 0.1$, yielding critical distance threshold $d_{\text{critical}} \approx 3.16$. The framework exhibits strong robustness to this choice (Section 4.2), as demonstrated by identical F_1-scores across $d_{\text{critical}} \in [1.414, 4.47]$. If uncertain, practitioners should use this default value without extensive tuning. The framework can handle false positive rates up to 24% without performance degradation due to effective self-filtering in the categorization stage.

Itemset Size Parameter (τ): Set $\tau = \lfloor |K|/2 \rfloor$ as the primary recommendation, with acceptable range $[\lfloor |K|/3 \rfloor, \lfloor 2|K|/3 \rfloor]$. This parameter critically affects performance (Section 4.3), with extreme values causing dramatic failures ($F_1 = 0.77$ for $\tau = 1$, $F_1 = 0.69$ for $\tau = 7$, compared to optimal $F_1 = 0.86$ for $\tau = 4$ when $|K| = 8$). The minimum value should be $\tau \geq 2$. The maximum value should be $\tau \leq \lfloor 3|K|/4 \rfloor$ to avoid excessive noise. When in doubt, prefer slightly lower τ over higher values.

Termination Parameters (ω, η, β): Use default values $\omega = 0.25$, $\eta = 0.1$, $\beta = 0.1$, as these performed consistently across all experiments. The parameter ω controls when to stop discovering new classes (when $|\mathcal{F}| < 0.25\times$ initial size). The parameter η sets minimum cluster size (discovered class must contain $\geq 10\%$ of $|\mathcal{F}|$). The parameter β defines refinement termination (stop when $< 10\%$ of $|\mathcal{F}|$ predicted as current class). Adjust η upward (0.15–0.20) if spurious small classes are discovered. Adjust β downward (0.05) if unknown classes have very uneven sizes.

Prediction Confidence Parameter (γ): Start with $\gamma = 0.6$ as the primary recommendation. For difficult datasets where iterative refinement stalls, reduce to $\gamma = 0.4$. For high-quality classifiers where more conservative labeling is desired, increase to $\gamma = 0.7$. Monitor the iterative refinement process in Algorithm 2 (lines 14–19); if fewer instances than expected are being incorporated, reduce γ.

Base Classifier Quality: The most critical requirement is ensuring base classifier quality exceeds 70% accuracy on known classes, with accuracy exceeding 90% strongly recommended (Section 4.4). Use cross-validation to assess base classifier quality before applying the framework. If accuracy falls below 70%, the framework will experience complete failure at the identification stage. Focus efforts on improving feature engineering, collecting more training data, or trying different classifier types before proceeding with the framework.

Number of Known Classes: Before applying the framework, consider the expected ratio $|U|/|K|$. Section 4.1 demonstrates that performance degrades substantially when $|U| \geq |K|$ (e.g., $F_1 = 0.65$ when $|U|/|K| = 1$ versus $F_1 = 0.87$ when $|U|/|K| = 0.25$). If this ratio approaches or exceeds 1, consider collecting more training data for additional known classes. The framework performs best when $|K| \geq |U|$, and is most reliable when $|U|/|K| \leq 0.5$.

Diagnostic Procedures: To assess whether the framework is working correctly on a new dataset, practitioners should monitor: (i) After Algorithm 1, check percentage flagged (typically 10–60%); if 0%, base classifier quality is likely insufficient or no unknown classes are present; if > 80%, possible issues with base classifier. (ii) During Algorithm 2, monitor number of classes discovered compared to expectations; check that discovered class sizes satisfy η threshold; examine iterative refinement (typically incorporates 10–40% additional instances per discovered class). (iii) After completion, evaluate confusion matrix for known classes to verify accuracy does not degrade > 10% from base classifier; check unknown class accuracy is reasonable (> 50% preferred).

When Not to Use: The sensitivity analysis reveals scenarios where the framework is inappropriate: (i) base classifier accuracy below 70%, (ii) ratio $|U|/|K| > 1$ without first increasing $|K|$, (iii) very few known classes ($|K| < 3$), (iv) expected micro-classes containing < 1% of dataset each (may not meet η threshold), and (v) true streaming scenarios requiring instant per-instance decisions.

5 Experiments and Results

This section evaluates the performance of the proposed methodology through four experiments, each conducted using datasets from diverse domains. The datasets are designed to test the framework's adaptability to various types of data, while the experiments assess its ability to detect and categorize unknown classes. The section is divided into two parts: the first describes the datasets, and the second details the experimental setups and results.

5.1 Data

The datasets are selected from diverse domains, including text, numeric, sensor, and image data, ensuring the methodology is robust and generalizable. Each dataset is vectorized into a specific number of features and varies in terms of known ($|\mathcal{K}|$) and unknown ($|\mathcal{U}|$) classes, as well as training (n_T) and open-world test (n_O) set sizes, as summarized in Table 10.

Table 10. Overview of the datasets used in the experiments

Experiment	Data	$\lvert\mathcal{K}\rvert$	$\lvert\mathcal{U}\rvert$	n_T	n_O
I	Code commit messages	5	0	1418	1351
II	Traditional numeric data	5	1	5416	2690
III	Human activity recognition	4	2	2898	2839
IV	Hand written digits	7	3	22504	6000

5.1.1 Code Commit Messages. The Code Commit Messages (CCM) dataset consists of 3,377 labeled messages sourced from open-source GitHub repositories (Nasir 2019). Each message belongs to one of five categories such as bug fixing, no category, design improvement, adding new features, and improving non-functional requirements. This dataset evaluates the framework's ability to classify text data, which is inherently noisy and diverse.

5.1.2 Traditional Numeric Data. The Traditional Numeric Data (TND) dataset includes 10,546 observations, derived from geo-spatial land cover classification (Johnson and Iizuka 2016). Each observation corresponds to one of six classes—impervious surfaces, farms, forests, grasslands, orchards, or water bodies. The dataset is represented by 29 features, including maximum normalized difference vegetation index values extracted from satellite imagery. It tests the framework's performance on structured, low-dimensional numeric data.

5.1.3 Human Activity Recognition Data. The HAR dataset contains 10,299 observations of human activities recorded via wearable sensors (Anguita et al. 2013). The dataset encompasses six activity classes, such as walking, sitting, and climbing stairs, each represented by 561 sensor-derived features. This dataset examines the methodology's ability to handle high-dimensional sensor data.

5.1.4 Handwritten Digits Data. The MNIST dataset is a standard benchmark for image classification tasks, consisting of 60,000 grayscale images of handwritten digits (0–9), each represented as a 784-dimensional vector (LeCun et al. 2010). This dataset is used to evaluate the framework's capability to classify high-dimensional visual data and detect novel digit classes.

5.2 Experiments

These experiments evaluate the framework's ability to handle open-world classification tasks under varying conditions. Each experiment focuses on datasets with an increasing number of unknown classes, ranging from 0 to 3. A consistent set of parameters is applied to ensure comparability across experiments. For each experiment, we use a random forest classifier. For each dataset, the accuracy, precision, recall, and F_1-score of the base classifier exceed 0.94.

The parameter settings for Experiments II, III, and IV follow the recommendations derived from comprehensive sensitivity analysis. Specifically, the itemset size parameter τ is set according to the guideline $\tau \approx |K|/2$, yielding $\tau = 2$ for Experiment I, $\tau = 3$ for Experiments II and III, and $\tau = 4$ for Experiment IV. The Chebyshev confidence parameter is set to $\alpha = 0.1$ (yielding $d_{\text{critical}} = 3.16$) across all experiments. The termination parameters use default values $\omega = 0.25$, $\eta = 0.1$, $\beta = 0.1$, and the prediction confidence $\gamma = 0.6$ (reduced to $\gamma = 0.4$ for Experiment IV due to increased complexity). Section 4 presents detailed sensitivity analysis examining the robustness of the framework to these parameters and identifying optimal configurations. The analysis demonstrates that the framework exhibits remarkable stability to variations in the Chebyshev parameter while requiring careful selection of the itemset size parameter and high-quality base classifiers with accuracy exceeding 70%.

5.2.1 Experiment I. This experiment validates that the framework does not hallucinate unknown classes when none exist. The training data contains all five classes present in the open-world data, creating a closed-world scenario. We apply Algorithm 1 to assess whether it incorrectly flags a substantial number of instances as unknown.

Algorithm 1 identified 52 out of 1,351 instances (3.8%) in the open-world data as potential anomalies. Table 11 shows the distribution of flagged instances across classes.

Based on Chebyshev's inequality with $\alpha = 0.1$, up to 10% of known-class instances may be flagged as potential unknowns due to natural variation in probability patterns. The observed 3.8% flag rate is well below this threshold, indicating that the framework correctly identifies this as a closed-world scenario. Since the flagged percentage (3.8%) is substantially less than the expected threshold (10%), and represents only 52 instances, Algorithm 2 would terminate immediately due to insufficient evidence of unknown classes (failing the η criterion). This validates the framework's ability to avoid false discoveries and demonstrates appropriate conservative behavior when no unknown classes are present.

Table 11. Experiment I: Distribution of Flagged Instances Across Classes

Class	Class Description	$\|O\|$	$\|\mathcal{F}\|$	% Flagged
1	Bug fixing	417	19	4.6
2	No category	117	1	0.9
3	Design improvement	181	8	4.4
4	Adding new features	229	5	2.2
5	Non-functional requirements	407	19	4.7
Total		**1,351**	**52**	**3.8**

5.2.2 Experiment II. The training data for this experiment contains classes *1, 2, 3, 4,* and *5*, while the open-world data includes one unknown class *0* alongside these known classes. As shown in Table 12, Algorithm 1 flags 1150 out of 2690 instances, with approximately 88% of these flagged instances correctly attributed to the unknown class *0*. This high identification rate indicates the framework's sensitivity to novel data.

Given the significant number of identified instances, the methodology progresses to the categorization stage, where Algorithm 2 successfully discovers and categorizes the unknown class. Table 14 shows that about 88% of the instances from class *0* are correctly classified. The remaining misclassified instances are predominantly labeled as class *1*, likely due to the similarity between class *0* (farm) and class *1* (forest).

Overall, the framework achieves a notable improvement in classifier performance, with accuracy increasing from 50% to 90%, as shown in Table 15. Additionally, precision, recall, and F_1-score exhibit corresponding increases, demonstrating the framework's efficacy in scenarios with a single unknown class. This experiment underscores the framework's ability to accurately identify and classify novel data while significantly improving predictive performance.

Table 12. Classwise distribution of O and $\mathcal{F}$ for TND dataset

Class	Class Information	$\|O\|$	$\|\mathcal{F}\|$	Percentage (%)
0	Farm	1224	1014	88.17
1	Forest	1230	91	7.90
2	Grass	126	15	1.73
3	Impervious	70	20	1.30
4	Orchard	8	4	0.52
5	Water	32	6	0.34

5.2.3 Experiment III. The open-world data here contains data from all classes, among which *1* and *5* are unknown, and the remainder are known. The classwise distribution of the data is displayed in Table 16. This table shows that 1990 out of 2839 instances in O are flagged, of which about 96% belong to unknown classes. After applying Algorithm 2, the two unknown classes are discovered, and the identified instances are assigned to two new classes.

We further evaluate the quality of this categorization using the confusion matrix in Table 17. Based on these results, nearly 90% of observations in class *1* are classified correctly, with most misclassifications going to class *2*, since class *1* (walking upstairs) is more similar to class *2* (walking downstairs). Similarly, 97.5% of observations in

Table 13. Confusion matrix for the initial performance of C on O for TND dataset

	Predicted class					
True class	0	1	2	3	4	5
0	0	1087	11	123	0	3
1	0	1126	0	4	0	0
2	0	6	57	7	0	0
3	0	13	1	112	0	0
4	0	7	0	0	1	0
5	0	4	0	3	0	25

Table 14. Confusion matrix for the final performance of C on O for TND dataset

	Predicted class					
True class	0	1	2	3	4	5
0	1082	109	1	29	0	3
1	72	1054	0	4	0	0
2	13	1	53	3	0	0
3	19	1	0	106	0	0
4	3	4	0	0	1	0
5	2	2	0	3	0	25

Table 15. Initial and final performance of C on O for TND dataset

Performance metric	Initial	Final
Accuracy	0.51	0.90
Precision	0.27	0.90
Recall	0.51	0.90
F_1-score	0.35	0.90

class *5* are correctly classified, with most remaining instances misclassified as class *3*, because class *5* (laying) is more similar to class *3* (sitting).

The overall accuracy of the classifier increases from 24% to 93% and the information regarding other performance metrics is provided in Table 18. This experiment demonstrates the successful performance of the proposed framework on data containing two unknown classes.

5.2.4 Experiment IV. The open-world data here contains observations belonging to digits *0* to *9*, with digits *1, 5,* and *8* as unknown classes and the remaining digits as known. Table 19 shows that about 86% of the instances in O are flagged as unknown, and more than 80% of these flagged instances belong to the unknown classes. Since the criteria to proceed further are met, we apply Algorithm 2 on $\mathcal{F}$. All three unknown classes are discovered, and the quality of this categorization is displayed in Table 20.

Table 16. Classwise distribution of O and $\mathcal{F}$ for HAR dataset

Class	Class information	$\lvert O\rvert$	$\lvert\mathcal{F}\rvert$	Percentage (%)
0	Walking	196	27	1.35
1	Walking upstairs	917	748	37.58
2	Walking downstairs	149	10	0.50
3	Sitting	175	13	0.65
4	Standing	211	30	1.50
5	Laying	1191	1162	58.39

Table 17. Confusion matrix for the final performance of C on O for HAR dataset

	Predicted class					
True class	1	2	3	4	5	6
0	167	21	0	0	0	8
1	2	822	92	0	1	0
2	7	9	133	0	0	0
3	0	0	0	162	13	0
4	0	7	0	10	194	0
5	0	1	0	34	2	1154

Table 18. Initial and final performance of C on O for HAR dataset

Performance metric	Initial	Final
Accuracy	0.24	0.93
Precision	0.09	0.94
Recall	0.24	0.93
F_1-score	0.12	0.93

About 79% of observations belonging to class *1*, 8% of those belonging to class *5*, and 73% of those belonging to class *8* are classified correctly. However performance on class *5* is poor compared to the other unknown classes, likely due to the lack of a common pattern among most observations in class *5*. Furthermore, only a few instances share any similarity pattern and are thus detected by Algorithm 2, leaving the classifier with insufficient observations to learn from.

For class *1*, most misclassifications are predicted as class *7*, reflecting similarity in digit shape. Similarly, digits *5* and *8* resemble digit *3*, causing the majority of misclassifications for these classes. Notably, the results support our hypothesis that the prediction patterns are informative and that the residual signature helps distinguish unknown classes. The final performance of the classifier increases from 48% to 75%, and other metrics are displayed in Table 21.

Table 19. Classwise distribution of O and $\mathcal{F}$ for MNIST data

Class	$\lvert O\rvert$	$\lvert\mathcal{F}\rvert$	Percentage (%)
0	435	59	1.71
1	1109	1104	32.09
2	396	78	2.26
3	433	81	2.35
4	425	98	2.84
5	892	790	22.96
6	422	76	2.20
7	454	79	2.29
8	999	988	28.72
9	435	87	2.52

Table 20. Confusion matrix for the final performance of C on O for MNIST data

	Predicted class									
True class	0	1	2	3	4	5	6	7	8	9
0	410	0	1	0	1	0	0	0	23	0
1	0	873	10	2	4	6	4	1	207	2
2	0	6	363	2	3	0	0	2	20	0
3	0	2	5	400	1	6	0	5	12	2
4	3	2	1	0	394	14	2	2	0	7
5	13	6	9	387	34	70	28	7	284	54
6	1	0	0	0	3	0	403	0	15	0
7	0	9	6	1	2	22	0	410	0	4
8	5	24	39	67	17	31	10	6	735	65
9	0	1	1	5	7	28	0	3	3	387

Table 21. Initial and final performance of C on O for MNIST data

Performance metric	Initial	Final
Accuracy	0.48	0.75
Precision	0.28	0.73
Recall	0.48	0.75
F_1-score	0.34	0.72

6 Case Study - Social Media Analytics for Community Resilience

Social media platforms such as Facebook and Twitter have become prevalent communication tools in modern society. These platforms provide a mechanism for collecting dynamic data on human behavior and sentiment. Such data has proven useful for studying a variety of activities, including crime prediction (Vivek and Prathap

2023), disease outbreaks (Zhang et al. 2025), stock market prices (Saravanaraj et al. 2025), and political election results (Gaur and Yadav 2025), among others.

Recent studies have explored the use of social media during natural disasters (Riccardi 2016), focusing on both the mood of the population and the various public reactions during specific incidents. Most of these works rely heavily on Twitter data for analysis (Reuter et al. 2018). One early study analyzed social media data during the 2008 wildfires in South Carolina (Sutton et al. 2008). Since then, case studies related to the Haiti earthquakes (Amiresmaili et al. 2021), Hurricane Ian (Karimiziarani and Moradkhani 2023), and Hurricane Laura (Zhou et al. 2023) have proliferated. Summaries of research on emergency management using Twitter can be found in (Luna and Pennock 2018; Martinez-Rojas et al. 2018), demonstrating the platform's importance as a social sensor with varying sensitivity to different disasters (Bhavaraju et al. 2019).

Social media analytics in disasters often involves sentiment analysis. Common sentiment classifications are binary (positive/negative) or three-way (positive/negative/neutral), while some studies perform more granular analyses (anger, disgust, fear, happiness, sadness, and surprise) (Schulz et al. 2013). For instance, (Mandel et al. 2012) conducted a demographic analysis of sentiment in Hurricane Irene tweets, and (R. J. Ragini et al. 2018) applied text mining on disaster-related tweets. Other researchers examined tweet sentiment during events like the 2013 Boston Marathon bombing (Lee et al. 2015), the 2017 Las Vegas shooting (Singh et al. 2018), the Syrian refugee crisis (Öztürk and Ayvaz 2018), and the COVID-19 outbreak (Jia et al. 2025). Beyond sentiment, works have classified disaster-related tweets into categories such as mitigation, preparedness, response, and recovery (Caragea, McNeese, et al. 2011; Q. Huang and Xiao 2015; Verma et al. 2011), or into "informative vs. non-informative" categories (Alam et al. 2018; Caragea, Silvescu, et al. 2016; J. R. Ragini et al. 2018) to assist emergency responders.

These studies underscore the role of social media analytics in disaster management and community resilience, aiming to extract useful information from tweets by classifying them into various informative or sentiment-based categories. Since the chosen categories are often manual and limited to the training data, it is possible that future tweets may not match any of these predefined categories. This scenario naturally creates an open-world classification problem, where the classifier may encounter unknown classes. The following demonstrates how open-world classification arises in Twitter analysis and evaluates our proposed framework on a real-world dataset.

We apply the proposed framework to the Crisis MMD dataset (Alam et al. 2018), which comprises human-annotated tweets collected during major disasters. The dataset under consideration contains 10,347 tweets corresponding to humanitarian categories such as affected individuals, infrastructure and utility damage, injured or dead people, and rescue volunteering or donation effort.

For this experiment, the class "rescue volunteering or donation effort" is designated as unknown, while the other classes are treated as known. To maintain consistency across analyses, we use a random forest classifier with TF–IDF-based word representations. Table 22 shows the distribution of the open-world data O and the flagged instances $\mathcal{F}$. About 30% of O is flagged, and 77% of these flagged instances actually belong to the unknown class. Since the flagged percentage exceeds 10%, we proceed to the second stage (Algorithm 2). In this experiment, We set Chebyshev's parameter $d_{\text{critical}} = 3.17$, the itemset size $\tau = 2$, and the termination parameters $\omega = 0.25$, $\beta = 0.1$, and $\eta = 0.1$. The prediction confidence γ is set to 0.6.

The algorithm automatically terminates after discovering a single unknown class in the open-world data. Table 24 shows that 82% of the instances in class *3* (the unknown) are correctly classified, with most of the remaining misclassifications falling under class *1*, this is consistent with the similarity between classes *1* and *3*. Before applying our framework, the initial classifier predictions on O (Table 23) demonstrated that most of the unknown-class tweets were misclassified as class *1*. The final results confirm that the framework discovered the unknown class and classified 1630 out of 1977 such tweets correctly. The overall accuracy of the classifier along with other performance metrics are provided in Table 24. Importantly, the performance on known classes did

not degrade substantially, illustrating the applicability of our approach to real-world social media analytics for improved community resilience.

Table 22. Classwise data distribution of O and $\mathcal{F}$ for Twitter data

Class	class information	$\lvert O\rvert$	$\lvert\mathcal{F}\rvert$
0	affected individuals	193	83
1	infrastructure and utility damage	479	57
2	injured or dead people	196	53
3	rescue volunteering or donation effort	1977	641

Table 23. Confusion matrix for the initial performance of C on O for Twitter data

	Predicted class			
True class	0	1	2	3
0	49	128	16	0
1	7	467	5	0
2	3	36	157	0
3	252	1631	94	0

Table 24. Confusion matrix for the final performance of C on O for Twitter data

	Predicted class			
True class	0	1	2	3
0	23	50	9	111
1	4	383	2	90
2	0	20	150	26
3	19	321	7	1630

Table 25. Initial and final performance of C on O for Twitter data

Performance metric	Initial	Final
Accuracy	0.24	0.77
Precision	0.08	0.79
Recall	0.24	0.77
F_1-score	0.11	0.76

7 Limitations and Future Work

While the proposed framework demonstrates effectiveness across diverse datasets and problem domains, several important limitations must be acknowledged based on the experimental results and sensitivity analysis.

The most critical limitation is the framework's requirement for high-quality base classifiers. As demonstrated in Section 4.4, the framework experiences complete failure when base classifier accuracy falls below approximately 70%, with optimal performance requiring accuracy exceeding 90%. This occurs because low-quality classifiers produce uncertain probability patterns for both known and unknown classes, eliminating the distinguishable difference that Algorithm 1 relies upon for anomaly detection. This restricts applicability to domains where high-accuracy classification is achievable.

Algorithm 1 may flag some known-class instances as false positives, particularly with lower critical distance thresholds. Section 4.2 shows that with $d_{\text{critical}} = 1.414$, approximately 24% of flagged instances were false positives. While Algorithm 2's categorization stage effectively filters many of these, as evidenced by consistent final performance ($F_1 = 0.81$) across different thresholds, some false positives persist. The trade-off between sensitivity and specificity is governed by the Chebyshev parameter α, which practitioners may need to adjust based on their specific cost considerations.

A significant limitation is the absence of direct comparisons to existing methods. This stems from our design philosophy of creating a model-agnostic framework operating on probability outputs rather than modifying specific algorithms. Most OWC methods are algorithm-specific. Direct comparison would require either re-implementing these methods in model-agnostic form or restricting our framework to specific classifiers, undermining our generality claims. Future work should conduct systematic benchmarking using the same base models as existing OWC methods, comparing performance on standard benchmarks, and analyzing trade-offs between model-agnostic generality and algorithm-specific optimizations.

The framework's performance depends critically on having sufficient known classes. Section 4.1 demonstrates that with only $|K| = 2$ known classes, the framework discovered only 1 of 2 unknown classes ($F_1 = 0.65$), failing to properly separate them. Performance improved substantially with $|K| = 5$ ($F_1 = 0.81$) and $|K| = 8$ ($F_1 = 0.87$). The framework performs best when $|K| \geq |U|$, and applications with high ratios of unknown to known classes should collect additional training data for more known classes if feasible.

Algorithm 2's sequential discovery process creates size bias, where larger unknown classes are discovered before smaller ones. Classes containing fewer than $\eta \times |\mathcal{F}|$ instances (default 10%) will not be discovered. Additionally, no backtracking mechanism exists; if the first discovered class incorrectly absorbs instances from multiple true unknown classes, this error propagates through subsequent iterations.

The itemset size parameter τ requires careful selection. Section 4.3 demonstrates that extreme values cause dramatic failures, with F_1-scores dropping from optimal 0.86 to 0.77 ($\tau = 1$) or 0.69 ($\tau = 7$). While the guideline $\tau \approx |K|/2$ works well across tested datasets, practitioners may need experimentation for domains with unusual characteristics. The framework provides no automated mechanism for parameter selection.

The framework assumes open-world data is available in batch form. Adapting to online settings where instances arrive sequentially would require substantial modifications, as itemset frequencies cannot be computed without sufficient instances, and the sequential discovery process requires iterating over the flagged set multiple times. While computationally efficient for tested datasets (up to 60,000 instances), scalability to millions of instances and hundreds of known classes has not been evaluated.

Future work should address these limitations by: (i) developing automated parameter selection mechanisms, (ii) exploring online adaptation strategies, (iii) conducting comprehensive baseline comparisons on standard benchmarks, (iv) investigating techniques to handle highly imbalanced unknown classes, and (v) extending the framework to truly large-scale applications with optimization strategies such as approximate itemset mining and incremental classifier updates.

8 Conclusion

To date, the open-world classification problem has been addressed primarily in the domain of computer vision. Most available approaches focus on image data and typically require data- or algorithm-specific adaptations. In contrast, this work aims to generalize open-world classification without restricting it to a particular domain or data type. To the best of our knowledge, no existing method in the literature solves open-world classification irrespective of data nature or classifier type. We address this gap by projecting the data onto a probabilistic space rather than the original feature space, enabling the methodology to be used with any classifier. Furthermore, while many existing techniques concentrate only on the identification of unknown instances, our framework completes the process by automatically categorizing these unknown observations. Some works have tried to estimate the number of unknown classes in the open-world data, but they do not accurately assign instances to distinct new classes. In contrast, we introduce the concept of a residual signature for unknown classes and leverage it to group unlabeled data. Our experiments show that the proposed approach works for different numbers of unknown classes and diverse data types, consistently increasing classifier performance while maintaining accuracy on known classes. The social media case study illustrates how the framework can be applied to real datasets for enhancing community resilience during disasters. Across the four experiments, the framework demonstrates substantial improvements: accuracy increases of 39 points (Experiment II: 0.51→0.90), 69 points (Experiment III: 0.24→0.93), 27 points (Experiment IV: 0.48→0.75), and 53 points (Twitter case study: 0.24→0.77), with final F_1-scores ranging from 0.72 to 0.93.

Acknowledgments

This research was partially funded by the National Institute of Standards and Technology (NIST) Center of Excellence for Risk-Based Community Resilience Planning through a cooperative agreement with Colorado State University [70NANB20H008 and 70NANB15H044].

References

R. Agrawal, T. Imieliński, and A. Swami. 1993. "Mining association rules between sets of items in large databases." In: *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, 207–216.

F. Alam, F. Ofli, and M. Imran. June 2018. "CrisisMMD: Multimodal Twitter Datasets from Natural Disasters." *Proceedings of the International AAAI Conference on Web and Social Media*, 12, 1, (June 2018). doi:10.1609/icwsm.v12i1.14983.

M. Amiresmaili, F. Zolala, M. Nekoei-Moghadam, S. Salavatian, M. Chashmyazdan, A. Soltani, and J. Savabi. 2021. "Role of social media in earthquake: A systematic review." *Iran. Red. Crescent Med. J*, 23, 5.

D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. 2013. "A public domain dataset for human activity recognition using smartphones." In: *ESANN*. Vol. 3, 3.

A. Bendale and T. Boult. June 2015. "Towards Open World Recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (June 2015).

A. Bendale and T. E. Boult. 2016. "Towards open set deep networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1563–1572.

S. K. T. Bhavaraju, C. Beyney, and C. Nicholson. 2019. "Quantitative analysis of social media sensitivity to natural disasters." *International Journal of Disaster Risk Reduction*, 39, 101251.

N. Burkart and M. F. Huber. 2021. "A survey on the explainability of supervised machine learning." *Journal of Artificial Intelligence Research*, 70, 245–317.

C. Caragea, N. McNeese, et al.. 2011. "Classifying text messages for the Haiti earthquake." In: *Proceedings of the 8th International ISCRAM Conference*. Lisbon, Portugal.

C. Caragea, A. Silvescu, and A. H. Tapia. 2016. "Identifying informative messages in disaster events using convolutional neural networks." In: *International Conference on Information Systems for Crisis Response and Management*, 137–147.

B. Fish and L. Reyzin. 2020. "On the complexity of learning a class ratio from unlabeled data." *Journal of Artificial Intelligence Research*, 69, 1333–1349.

A. Gaur and D. K. Yadav. 2025. "A comprehensive analysis of forecasting elections using social media text." *Multimedia Tools and Applications*. Advance online publication. doi:10.1007/s11042-024-20528-w.

Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi. 2017. *Generative OpenMax for Multi-Class Open Set Classification.* arXiv preprint arXiv:1707.07418. (2017). https://arxiv.org/abs/1707.07418 arXiv: 1707.07418 (cs.CV).

C. Geng and S. Chen. 2022. "Collective Decision for Open Set Recognition." *IEEE Transactions on Knowledge and Data Engineering*, 34, 1, 192–204.

C. Geng, S.-J. Huang, and S. Chen. 2021. "Recent Advances in Open Set Recognition: A Survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43, 10, 3614–3631. doi:10.1109/TPAMI.2020.2981604.

N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld. 2013. "Toward supervised anomaly detection." *Journal of Artificial Intelligence Research*, 46, 235–262.

J. Guo, H. Wang, Y. Xu, W. Xu, Y. Zhan, Y. Sun, and S. Guo. 2025. "Multimodal Dual-Embedding Networks for Malware Open-Set Recognition." *IEEE Transactions on Neural Networks and Learning Systems*, 36, 3, 4545–4559. doi:10.1109/TNNLS.2024.3373809.

K. Han, S.-A. Rebuffi, S. Ehrhardt, A. Vedaldi, and A. Zisserman. 2021. "Autonovel: Automatically discovering and learning novel visual categories." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44, 10, 6767–6781.

M. Hassen and P. K. Chan. 2020. "Learning a neural-network-based representation for open set recognition." In: *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 154–162.

Q. Huang and Y. Xiao. 2015. "Geographic situational awareness: mining tweets for disaster preparedness, emergency response, impact, and recovery." *ISPRS International Journal of Geo-Information*, 4, 3, 1549–1568.

L. P. Jain, W. J. Scheirer, and T. E. Boult. 2014. "Multi-class open set recognition using probability of inclusion." In: *European Conference on Computer Vision*. Springer, 393–409.

B. Jia, M. Xie, J. Wu, and J. Zhao. 2025. "Underneath social media texts: Sentiment responses to public health emergency during 2022 COVID-19 pandemic in China." *International Journal of Disaster Risk Reduction*, 118, 105239. doi:10.1016/j.ijdrr.2025.105239.

B. A. Johnson and K. Iizuka. 2016. "Integrating OpenStreetMap crowdsourced data and Landsat time-series imagery for rapid land use/land cover (LULC) mapping: Case study of the Laguna de Bay area of the Philippines." *Applied Geography*, 67, 140–149.

K. J. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian. 2021. "Towards Open World Object Detection." In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5826–5836. doi:10.1109/CVPR46437.2021.00577.

P. R. M. Júnior, R. M. De Souza, R. d. O. Werneck, B. V. Stein, D. V. Pazinato, W. R. de Almeida, O. A. Penatti, R. d. S. Torres, and A. Rocha. 2017. "Nearest neighbors distance ratio open-set classifier." *Machine Learning*, 106, 3, 359–386.

N. Kardan and K. O. Stanley. 2017. "Mitigating fooling with competitive overcomplete output layer neural networks." In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 518–525.

M. Karimiziarani and H. Moradkhani. 2023. "Social response and Disaster management: Insights from twitter data Assimilation on Hurricane Ian." *International journal of disaster risk reduction*, 95, 103865.

M. Kaur and S. Kang. 2016. "Market Basket Analysis: Identify the changing trends of market data using association rule mining." *Procedia Computer Science*, 85, 78–85.

S. Kullback and R. A. Leibler. 1951. "On Information and Sufficiency." *The Annals of Mathematical Statistics*, 22, 1, 79–86.

Y. LeCun, C. Cortes, and C. Burges. 2010. "MNIST handwritten digit database." *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

J. Lee, B. A. Rehman, M. Agrawal, and H. R. Rao. 2015. "Sentiment analysis of Twitter users over time: the case of the Boston bombing tragedy." In: *Workshop on E-Business*. Springer, 1–14.

R. Li, D. Zhang, Y. Wang, Y. Jiang, Z. Zheng, S.-W. Jeon, and H. Wang. 2025. "Open-Vocabulary Multi-Object Tracking With Domain Generalized and Temporally Adaptive Features." *IEEE Transactions on Multimedia*, 27, 3009–3022. doi:10.1109/TMM.2025.3557619.

S. Li, T. Fischer, L. Ke, H. Ding, M. Danelljan, and F. Yu. June 2023. "OVTrack: Open-Vocabulary Multiple Object Tracking." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (June 2023), 5567–5577.

J. Lin. 1991. "Divergence measures based on the Shannon entropy." *IEEE Transactions on Information Theory*, 37, 1, 145–151.

S. Luna and M. J. Pennock. 2018. "Social media applications and emergency management: A literature review and research agenda." *International Journal of Disaster Risk Reduction*, 28, 565–577.

B. Mandel, A. Culotta, J. Boulahanis, D. Stark, B. Lewis, and J. Rodrigue. 2012. "A demographic analysis of online sentiment during hurricane Irene." In: *Proceedings of the Second Workshop on Language in Social Media*. Association for Computational Linguistics, 27–36.

M. Martinez-Rojas, M. del Carmen Pardo-Ferreira, and J. C. Rubio-Romero. 2018. "Twitter as a tool for the management and analysis of emergency situations: A systematic literature review." *International Journal of Information Management*, 43, 196–208.

M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. Van De Weijer. 2022. "Class-incremental learning: survey and performance evaluation on image classification." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 5, 5513–5533.

T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. 2013. "Distance-based image classification: Generalizing to new classes at near-zero cost." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 11, 2624–2637.

Nasir. 2019. *Multi-Class Text Classification – Random Forest.* https://github.com/nxs5899/Multi-Class-Text-Classification----Random-Forest. Accessed: 2025-02-02. (2019).

L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li. 2018. "Open set learning with counterfactual images." In: *Proceedings of the European Conference on Computer Vision (ECCV)*, 613–628.
P. Oza and V. M. Patel. 2019. "C2ae: Class conditioned auto-encoder for open-set recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2307–2316.
N. Öztürk and S. Ayvaz. 2018. "Sentiment analysis on Twitter: A text mining approach to the Syrian refugee crisis." *Telematics and Informatics*, 35, 1, 136–147.
G. Pang, K. M. Ting, D. Albrecht, and H. Jin. 2016. "Zero++: Harnessing the power of zero appearances to detect anomalies in large-scale data sets." *Journal of Artificial Intelligence Research*, 57, 593–620.
P. Patel, B. Sivaiah, R. Patel, and R. Choudhary. 2024. "Association Rule Mining for Healthcare Data Analysis." In: *Computational Intelligence in Healthcare Informatics*. Springer, 127–139.
D. S. Prijatelj, S. Grieggs, J. Huang, D. Du, A. Shringi, C. Funk, A. Kaufman, E. Robertson, and W. J. Scheirer. 2024. "Human Activity Recognition in an Open World." *Journal of Artificial Intelligence Research*, 81, 935–971.
H. Qu, X. Hui, Y. Cai, and J. Liu. 2024. "Lmc: Large model collaboration with cross-assessment for training-free open-set object recognition." *Advances in Neural Information Processing Systems*, 36.
J. R. Ragini, P. R. Anand, and V. Bhaskar. 2018. "Mining crisis information: A strategic approach for detection of people at risk through social media analysis." *International Journal of Disaster Risk Reduction*, 27, 556–566.
R. J. Ragini, R. P. Anand, and V. Bhaskar. 2018. "Big data analytics for disaster response and recovery through sentiment." *International Journal of Information Management*, 42, 13–24.
S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. 2017. "iCaRL: Incremental Classifier and Representation Learning." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5533–5542.
T. Ren et al.. 2025. *DINO-X: A Unified Vision Model for Open-World Object Detection and Understanding*. (2025). https://arxiv.org/abs/2411.14347 arXiv: 2411.14347 (cs.CV).
C. Reuter, G. Backfried, M.-A. Kaufhold, and F. Spahr. 2018. "ISCRAM turns 15: A Trend Analysis of Social Media Papers 2004-2017." In: *Proceedings of the 15th ISCRAM conference*.
M. T. Riccardi. 2016. "The power of crowdsourcing in disaster response operations." *International Journal of Disaster Risk Reduction*, 20, 123–128.
M. Ristin, M. Guillaumin, J. Gall, and L. Van Gool. 2014. "Incremental learning of NCM forests for large-scale image classification." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3654–3661.
S. Saravanaraj, V. Govindan, S. Broumi, and H. Byeon. 2025. "Sentimental Analysis to Predict Stock Market Using in Neutrosophic Time Series." *International Journal of Neutrosophic Science (IJNS)*, 25, 2.
J. G. Saw, M. C. Yang, and T. C. Mo. 1984. "Chebyshev inequality with estimated mean and variance." *The American Statistician*, 38, 2, 130–132.
W. J. Scheirer, L. P. Jain, and T. E. Boult. 2014. "Probability models for open set recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 11, 2317–2324.
W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. 2013. "Toward open set recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 7, 1757–1772.
M. D. Scherreik and B. D. Rigling. 2016. "Open set recognition for automatic target classification with rejection." *IEEE Transactions on Aerospace and Electronic Systems*, 52, 2, 632–642.
A. Schulz, T. D. Thanh, H. Paulheim, and I. Schweizer. 2013. "A fine-grained sentiment analysis approach for detecting crisis related microposts." In: *ISCRAM*.
M. Shahin, S. Arakkal Peious, R. Sharma, M. Kaushik, S. Ben Yahia, S. A. Shah, and D. Draheim. 2021. "Big data analytics in association rule mining: A systematic literature review." In: *2021 the 3rd International Conference on Big Data Engineering and Technology (BDET)*, 40–49.
X. Shi, Y. Zhao, and H. Du. 2024. "A data mining method for biomedical literature based on association rules algorithm." *International Journal of Data Mining and Bioinformatics*, 28, 1, 1–17.
L. Shu, H. Xu, and B. Liu. 2017. *DOC: Deep Open Classification of Text Documents*. arXiv preprint arXiv:1709.08716. (2017). https://arxiv.org/abs/1709.08716 arXiv: 1709.08716 (cs.CL).
L. Shu, H. Xu, and B. Liu. 2018. *Unseen Class Discovery in Open-world Classification*. arXiv preprint arXiv:1801.05609. (2018). https://arxiv.org/abs/1801.05609 arXiv: 1801.05609 (cs.LG).
N. Singh, N. Roy, and A. Gangopadhyay. 2018. "Analyzing the Sentiment of Crowd for Improving the Emergency Response Services." In: *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 1–8.
J. Sutton, L. Palen, and I. Shklovski. May 2008. "Backchannels on the front lines: Emergent uses of social media in the 2007 southern California wildfires." In: *Proceedings of the 5th International ISCRAM Conference* (Washington, DC, United States). Washington, DC, United States, (May 2008), 624–632.
A. Toumpa and A. G. Cohn. 2023. "Object-agnostic affordance categorization via unsupervised learning of graph embeddings." *Journal of Artificial Intelligence Research*, 77, 1–38.

S. Vaze, K. Han, A. Vedaldi, and A. Zisserman. 2022. “Open-Set Recognition: A Good Closed-Set Classifier is All You Need.” In: *International Conference on Learning Representations*.

S. Verma, S. Vieweg, W. J. Corvey, L. Palen, J. H. Martin, M. Palmer, A. Schram, and K. M. Anderson. 2011. “Natural language processing to the rescue? extracting" situational awareness" tweets during mass emergency.” In: *Fifth International AAAI Conference on Weblogs and Social Media*.

M. Vivek and B. R. Prathap. 2023. “Spatio-temporal crime analysis and forecasting on twitter data using machine learning algorithms.” *SN Computer Science*, 4, 4, 383.

J. Wu et al.. 2024. “Towards Open Vocabulary Learning: A Survey.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46, 7, 5092–5113. doi:10.1109/TPAMI.2024.3361862.

Q. Yan, Y. Yang, Y. Dai, X. Zhang, K. Wiltos, M. Woźniak, W. Dong, and Y. Zhang. 2025. “CLIP-guided continual novel class discovery.” *Knowledge-Based Systems*, 310, 112920. doi:https://doi.org/10.1016/j.knosys.2024.112920.

H.-M. Yang, X.-Y. Zhang, F. Yin, Q. Yang, and C.-L. Liu. 2020. “Convolutional prototype network for open set recognition.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44, 5, 2358–2370.

H. Zhang, C. Yang, X. Deng, and C. Luo. 2025. “How Authoritative Media and Personal Social Media Influence Policy Compliance Through Trust in Government and Risk Perception: Quantitative Cross-Sectional Survey Study.” *Journal of Medical Internet Research*, 27, e64940.

X. Zhong and J. Cui. 2025. “Positive–negative prototypes fusion framework for open set recognition.” *Scientific Reports*, 15, 1, 23815.

S. Zhou, P. Kan, Q. Huang, and J. Silbernagel. 2023. “A guided latent Dirichlet allocation approach to investigate real-time latent topics of Twitter data during Hurricane Laura.” *Journal of Information Science*, 49, 2, 465–479.

F. Zhu, Z. Cheng, X.-Y. Zhang, and C.-L. Liu. June 2023. “OpenMix: Exploring Outlier Samples for Misclassification Detection.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (June 2023), 12074–12083.

F. Zhu, S. Ma, Z. Cheng, X.-Y. Zhang, Z. Zhang, and C.-L. Liu. 2024. *Open-world Machine Learning: A Review and New Outlooks*. arXiv preprint arXiv:2301.12345. (2024). arXiv: 2403.01759 (cs.LG).

Received 11 April 2025; accepted 13 January 2026