Transductive Transfer Learning for Nuclear Reactor Monitoring

A. Hashemizadeh[a], B.L. Goldblum[b,*], J. E. Bevins[c], M. Brinker[d], M. Mathew[a], A. Patel[a], S. Sriram[a], C.L. Stewart[a], J. Tibbetts[a], J. Whetzel[d]

[a]*Department of Nuclear Engineering, University of California Berkeley, Berkeley CA USA*
[b]*Nuclear Science Division, Lawrence Berkeley National Laboratory, Berkeley CA USA*
[c]*Department of Engineering Physics, Air Force Institute of Technology, Wright-Patterson AFB, OH USA*
[d]*Sandia National Laboratories, LLC, Albuquerque, NM USA*

Abstract

Machine learning models crafted using data obtained in a specific setting often produce incorrect outputs when applied to data collected in a different setting—even for the same classification task. Transferability, the application of models generated in one setting to other contexts or settings, is critical to the application of machine learning in proliferation detection scenarios to enable monitoring of a wide range of nuclear facilities of interest. This work presents a new transductive transfer learning technique, where a source model is generated using labeled data obtained from a testbed facility and then adapted via incremental learning with proxy-labeled data from the target domain obtained using a cluster-then-label approach. To demonstrate this method for classifying the operational state of a nuclear reactor, data were collected at two different reactor facilities: the High Flux Isotope Reactor at Oak Ridge National Laboratory, which served as the testbed facility, and a TRIGA reactor at the McClellan Nuclear Research Center, which represented the transfer target. The cluster-then-label approach yielded proxy labels with >88% accuracy, and the target model yielded an increase in classification accuracy of almost 10% relative to the performance of the model trained using only data from the source domain. This cluster-then-label approach to transferability is a potential solution to a presently unmet need in the proliferation detection community for generalizing the applicability of testbed-generated multisource datasets to real-world applications.

Keywords: supervised classification, multimodal analytics, transfer learning, clustering

1. Introduction

Detection, identification, and attribution of illicit nuclear materials and related technologies has been a long-stated goal of the Treaty on the Non-Proliferation of Nuclear Weapons (NPT) [1]. To ensure the security challenges arising from nuclear weapons are met, state parties to the NPT conclude agreements with the International Atomic Energy Agency (IAEA) wherein fissile material production is monitored via nuclear materials accountancy and inspection measures [2]. Current technical detection capabilities for undeclared activities generally rely upon nuclear forensics, defined by the IAEA as "the examination of nuclear or other radioactive material, or of evidence that is contaminated with radionuclides, in the context of legal proceedings under international or national law related to nuclear security [3]." Unlike most scientific disciplines, nuclear forensics does not generally deal with well-defined samples. Samples may consist of almost any substance, from gases to microscopic particles to human bodily excretions [4]. While nuclear forensic techniques provide robust methods of detecting the presence of nuclear materials, sample collection can be very difficult in non-permissive applications, and the operational history of the facility from which a sample was derived may be obscured due to confounding factors. As such, new detection and characterization approaches are needed to meet international nonproliferation objectives.

Multisource machine learning using nonradiological data has high potential for applications in nuclear reactor monitoring to address nonprolifera-

*Corresponding author
Email address: bethany@lbl.gov (B.L. Goldblum)

tion detection challenges. Multisource sensors can monitor physical, material, electromagnetic, and pattern-of-life signatures that can be used to assess the operational state of a nuclear facility [5]. The use of multiple sensors potentially allows for a more robust determination of the operational status of a nuclear reactor than is afforded by materials analysis alone. In addition, an array of multisensors could be deployed to optimize the collection of signals over a region for area monitoring. This opens the potential to use correlated signatures of interrelated phenomena, e.g., fuel delivery, start-up activities, thermal discharge, etc. to classify nuclear reactor operations.

Supervised classification is a powerful machine learning technique and has been shown to yield high classification accuracy for a wide range of tasks [6]. However, it requires a training set of labeled data, i.e., "ground truth" derived from detailed operational logs or on-the-ground observations, which may be difficult to obtain for some nuclear facilities of interest. Unsupervised classification obviates the requirement for ground truth, but classification tasks can be more error prone for specific problems in comparison to supervised approaches [7]. Testbed facilities allow for the development and validation of supervised machine learning methods, providing a sandbox for new multimodal analyses and illuminating the art of the possible. However, the challenge remains to apply these classifiers in new settings—to enable the monitoring of operations at a variety of nuclear facilities of interest.

Transfer learning is the process of applying a model developed for one task or setting to another task or setting [8]. Traditionally, inductive approaches have been used [9], where a source model is trained on data obtained in the source setting, for example the testbed facility, and then applied to the data obtained in another setting, the target nuclear reactor of interest. Such an approach makes an implicit assumption that the feature-space distribution is the same (or sufficiently similar) in both settings. However, individual nuclear facilities differ significantly in terms of their peak power production, fuel inventory, coolant type, reactor vessel and core characteristics, reactivity control, coolant and containment systems, etc., and these changes will manifest as changes in the underlying feature-space distributions.

Transductive transfer learning methods avoid this assumption by using information embedded in the target dataset to develop a modified model [10]. By leveraging information, e.g., model type and architecture, features, parameter weights, etc., from a source model previously trained on testbed data, reasoning from training instances can be applied in a setting where no labeled data are available. Transductive transfer learning has shown success in real-world applications including action recognition, image classification, and computer vision [11, 12, 13], but little to no work has been performed to assess its applicability and performance for multimodal supervised classification for proliferation detection scenarios.

In this work, a new approach to transductive transfer learning is demonstrated using datasets obtained from two different nuclear reactor facilities. With labeled data from the testbed reactor, a feed-forward neural network—the source model—was trained to classify the reactor operational state. A cluster-then-label approach was applied to data from both the testbed and target facilities to generate proxy labels for the target dataset. The source model was then further trained using the proxy-labeled target data to yield improved classification performance in the target domain. Section 2 describes the multisensor platform, the nuclear reactor facilities, and the data collection campaigns. Section 4 provides an overview of MIMOSAS, the software library used to execute the machine learning workflow. In Section 5, the transductive transfer learning method is described along with details related to data preprocessing and the proxy-labeling procedure. The results of the cluster-then-label approach for transfer learning are provided in Section 6. Concluding remarks are given in Section 7.

2. Data

Data collection was performed at two nuclear reactors: the High Flux Isotope Reactor at Oak Ridge National Laboratory, which served as the testbed facility, and a TRIGA research reactor at the McClellan Nuclear Research Center, which represented the target facility. In this section, the multisensor platform and associated data products are described. Then, details on each of the nuclear reactors and experimental campaigns are given.

2.1. Merlyn Multisensor

Data were collected using the Merlyn multisensor platform designed by Special Technologies

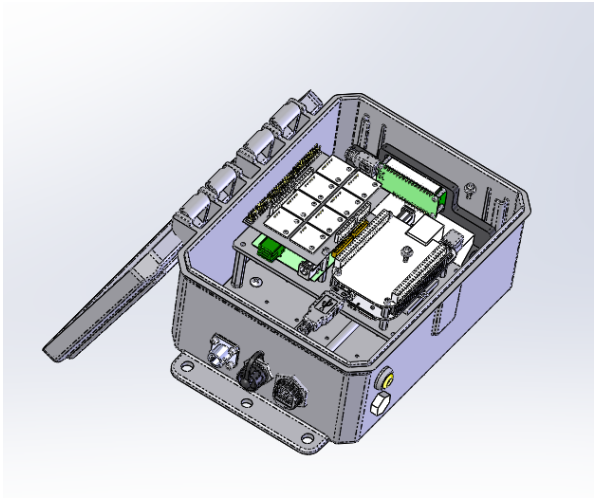


Figure 1: Schematic of Merlyn multisensor platform showing board arrangement, connections, and protective enclosure.

Table 1: Merlyn sensor suite.

Modality	Sensor
Acceleration (3-axis)	Kionix KX-224-1053 [17]
Ambient Light	ROHM RPR-0521RS [18]
Color (RGB)	ROHM BH1745NUC [19]
GPS	Adafruit Ultimate GPS Featherwing [20]
Magnetic Field (3-axis)	ROHM 1422AGMV [21]
Pressure (Barometric)	ROHM BM1383AGLV [22]
Proximity	ROHM RPR-0521RS [18]
Temperature	ROHM BD1020HFV [23]

Laboratory, an organization within the Nevada National Security Site complex of facilities. A schematic of the Merlyn is shown in Figure 1. It is comprised of a BeagleBone Black mainboard [14], an ATmega328P-based Arduino UNO breakout board [15], a ROHM SensorShield EVK-003 sensor package [16], and supporting hardware to distribute power and data among these components. The sensor package onboard each Merlyn is listed in Table 1. While the GPS module is primarily used to establish system clock time, the remaining modalities are used as inputs to the machine learning models developed in this work. Each sensor samples at 16 Hz, and the output is written to an onboard microSD card. Data are retrieved either via ethernet or USB connection. The sensor package was designed to sample signals across multiple domains including personnel or vehicle movements, local changes in the ambient environment, and operation of large-scale experimental and support apparatus. The Merlyn is housed in an environmental enclosure with ports for power, USB, and RJ45 connections, as well as a passthrough for pressure equalization.

2.2. High Flux Isotope Reactor

The High Flux Isotope Reactor (HFIR) at Oak Ridge National Laboratory is an 85 MW, HEU-fueled research reactor and is co-located with the Radiochemical Engineering Development Center (REDC), which performs isotope production research and irradiation target processing. In May 2019, an array of 12 Merlyns was fielded across established data collection nodes at the HFIR/REDC complex in cooperation with the Multi-Informatics for Nuclear Operations Scenarios (MINOS) venture [24]. This array, depicted in Figure 2, spans buildings that house the main research equipment (reactor and target assembly/disassembly hot cells); supporting structures and equipment such as heat rejection, fluid transport machinery, and climate control; and personnel work areas. Data collected by the multisensor network array are automatically pushed on an hourly basis to a dedicated data server managed by the MINOS venture.

Extensive ground truth on HFIR/REDC operations are available via the MINOS venture, with metadata in the form of two primary categories: operational and parametric. Operational ground truth provides logs for high-level operations phenomena such as target production, material transfers, etc. Parametric data includes a wide range of detailed metrics derived from operational diagnostics such as fan speed, coolant temperature, reactor power, etc. To generate labels for reactor operational status, the parametric data for the averaged heat output of the reactor core was binned into categories based on fraction of nominal full power. A threshold of 10% of full power was applied to assign operational status. That is, below this threshold, the reactor status was considered “off” for the binary classification problem; at or exceeding this threshold indicates “reactor on”. A threshold of 10% of full power was adopted as the boundary between the “reactor off” and “reactor on” categories given that the lowest partial power hold in

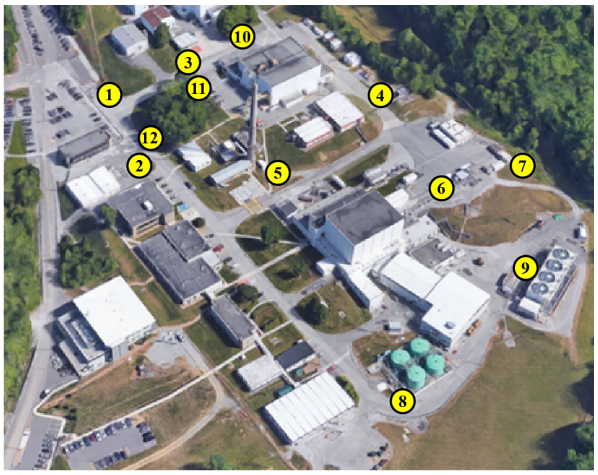


Figure 2: Overhead image of the High Flux Isotope Reactor at Oak Ridge National Laboratory. The numbered yellow circles denote the positions of the Merlyn multisensors.

the HFIR startup procedure is at 10%.

2.3. McClellan Nuclear Research Center

The McClellan Nuclear Research Center (MNRC) houses a 1 MW TRIGA reactor used for nuclear research and applications, including component and hardware imaging and target irradiation. Five Merlyns were deployed in July 2020 at the MNRC, at locations shown in Figure 3. The Merlyns were positioned near equipment and structures similar to those present at the MINOS nodes: the personnel and vehicle entrances, the reactor heat rejection system, the auxiliary electrical/support systems, and the heavy cranes.

Operations logs were retrieved and data were labeled for use in transductive transfer learning method development and testing. As the MRNC operates during regular business hours, a full startup and shutdown cycle is completed on a daily basis. Operation logs provided the following information for each cycle: date, startup time, shutdown time, reactivity excess, run duration, megawatt-hours per run, and average power level. To generate labels for the Merlyn data, the power output recorded in the logs was converted into fractional nominal power and split according to the same 10% threshold used for labeling the HFIR power status. Unlike HFIR, the TRIGA startup procedure only has two holds: one at mW-scale and one at ∼50 W, both well below the level of heat generation necessitating active cooling. They will thus appear from outside containment as if the reactor is off—an indication that the 10% threshold cleaves operations at both facilities reasonably well.

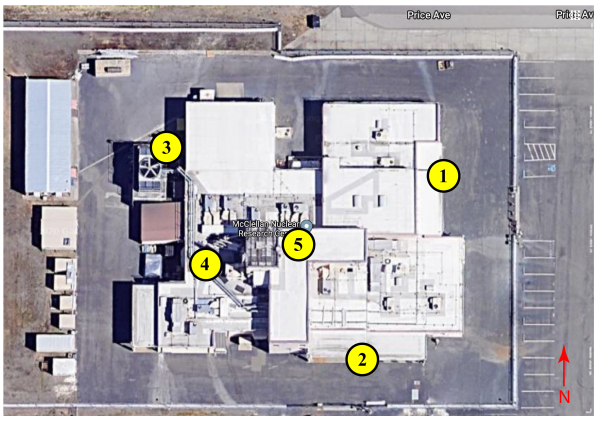


Figure 3: Merlyn multisensor array locations at the McClellan Nuclear Research Center.

2.4. Management, Preservation, and Selection

Data from each multisensor network were transferred to a local server on a monthly basis. A suite of custom quality assurance scripts were executed to ensure data integrity. These check for and remove incomplete entries and nonphysical or otherwise out-of-range sensor readings and provide the analyst with time-series and distribution plots on both the readings themselves and on device-specific and array-wide sensor uptime. Plots of the data were further inspected for nonphysical data indicative of possible sensor malfunction. Upon passing the quality checks, data were packaged into HDF5 containers and transferred for preservation on the MINOS database [25] and Berkeley Data Cloud [26] for the HFIR/REDC and MNRC data, respectively.

The supervised classification and tranductive transfer learning demonstration presented in this work use data from a single Merlyn multisensor at each nuclear reactor facility (as opposed to data from the full multisensor array). Each multisensor at a given facility has a different position relative to the main components of the nuclear reactor and thus the measured sample distributions vary by node. To mitigate the potential for bias introduced by these disparate distributions as well as to prioritize "similar" nodes located near equipment essential to operation of the reactor, data from multisensors positioned closest to the cooling tower at each facility were used. For the HFIR/REDC source domain, data from HFIR-09 (i.e., Merlyn 9

in Figure 2) were used, and for the MNRC target domain, data from MNRC-03 (i.e., Merlyn 3 in Figure 3) were used.

3. Domain Adaptation using Cluster-then-label Feature Mapping

Domain adaptation is a type of transfer learning where a model trained on data from a source domain is adapted for improved performance on a target domain. If the target domain data are labeled, this can be accomplished using incremental learning, whereby the source domain model parameters are transferred wholly or partially to a target model, which is then further trained using data from the target domain to achieve the desired classification performance [27]. In the case of unlabeled target data, unsupervised domain adaptation using backpropagation was introduced for feed-forward neural network model architectures, whereby classification decisions are made based on features with the same or similar distributions in the source and target domains [28]. In this work, cluster-then-label feature mapping is introduced as a new means by which domain adaptation may be mediated for unlabeled target domains.

The cluster-then-label feature mapping approach to facilitate domain adaptation is an extension of the cluster-then-label semi-supervised learning approach introduced by Peikari et al. [29]. Peikari et al. applied clustering analysis to both labeled and unlabeled data from the same domain and then assigned labels to each unlabeled point based upon the label of the closest labeled point as determined using Euclidean distance. To apply the cluster-then-label approach for domain adaptation, a density-based clustering algorithm is used to identify the shared underlying structure of the data spaces of both the source and target domains *in a way that distinguishes the label space*. The relative positions of clusters with respect to the origin of the particular domain data space are used to establish a one-to-one mapping between the source and target clusters. The (known) label distributions of the source domain clusters are then applied to their paired target domain clusters as proxy labels.

Proxy-labeled data from the target set are used to refine the source model through an incremental learning step. This is done using two approaches: continual learning and "learning without forgetting," in which training is continued while freeing only the weights of select layers of the source network [30]. The outcome is a tailored classifier that takes advantage of the information obtained from the source setting, yielding a high performance machine learning model honed for the target domain without the requirement of labeled target data.

4. Methods

The machine learning models and supporting algorithms used in this work were developed using an extension of the Multisource Input Model Output Security Analysis Suite (MIMOSAS) [31]. MIMOSAS is a software suite for machine learning classification of multisource data for nuclear security applications. The first released version of MIMOSAS was constructed as a self-contained application for binary classification of multisource data with a variety of basic classifier options (i.e., decision tree, random forest, and feed-forward neural network). The release also included preprocessing tools, hyperparameter optimization support, and performance evaluation metrics. To incorporate capabilities for transductive transfer learning and time-series classification, MIMOSAS is currently under development as a library encompassing a wide range of helper functions, tools for dealing with large data sets, and other resources. MIMOSAS leverages various functions and objects from the following Python libraries: TensorFlow [32], NumPy [33], SciPy [34], scikit-learn [35], and h5py [36]. The beta development branch of MIMOSAS was used for the analysis presented herein.

4.1. Preprocessing

To prepare the Merlyn sensor data for source model training, MIMOSAS offers a number of data preprocessing functions. The pressure and temperature data streams from the testbed dataset were background corrected by subtracting ambient weather conditions obtained from data collected by a nearby weather station managed by the National Oceanic and Atmospheric Administration (NOAA) [37]. Then, the x, y, and z components of the acceleration and magnetic field measurements were either left unaltered (i.e., ingested as individual components) or combined by taking the ℓ^2-norm (i.e., Euclidean norm) of the individual vector components to obtain the acceleration and magnetic field magnitudes, $|\mathbf{m}|$:

$$|\mathbf{m}| = \sqrt{x^2 + y^2 + z^2}. \tag{1}$$

For each data stream except for proximity, the z-score, z_i, of each data point, x_i, was then obtained using standardization:

$$z_i = \frac{x_i - \mu}{\sigma}, \tag{2}$$

where μ is the mean and σ is the standard deviation of the distribution of measurements of a given modality. As the proximity data were largely single-valued events approximating a binary variable representative of the presence or absence of an object, they were instead scaled to the interval [0, 1]:

$$p_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}, \tag{3}$$

where $\max(x)$ and $\min(x)$ represent the maximum and minimum values of the distribution of proximity measurements, respectively. Finally, labels were assigned on a per sample basis denoting the reactor operational state using the ground truth described in Sec. 2. Each sample was comprised of a measurement obtained at a given time for all sensing modalities listed in Table 1 (except GPS).

The same preprocessing steps were applied to the data from the source and target domains. For the cluster-then-label procedure, the variance of each modality was calculated for a series of 60-minute time segments. Min-max scaling was then applied to the variance distributions on a per-domain basis.

4.2. Data Ingestion

A streaming approach to data training and evaluation was adopted in MIMOSAS to facilitate scalability and mitigate hardware memory limitations when dealing with large datasets. A data generator was constructed to load data in manageable batch sizes. These batches were then cached and saved for more efficient data streaming at later epochs or for use in other models leveraging the same data.

4.3. Model Architecture

A multilayer feed-forward neural network was adopted as the classifier in a supervised learning approach. A feed-forward neural network is a directed computational graph of nodes arranged into layers; at each layer, values from the incoming edges are weighted, summed, and biased before being passed into an activation function whose output value is passed to nodes in the next layer. While a feed-forward network can approximate an arbitrary function with one hidden layer consisting of a large number of nodes, the addition of a second hidden layer enables the introduction of nonlinearities by nesting activation functions to achieve a given approximation fidelity with a smaller total number of hidden nodes. Dropout regularization, which removes a fraction of randomly-selected nodes in each training epoch, was applied in each hidden layer to combat overfitting;[39] during prediction, dropout layers are inactive. The dropout rate, or the probability during each training step that any given node is removed from the calculation (i.e., a dropout rate of 0 means there is no dropout), was treated as a hyperparameter.

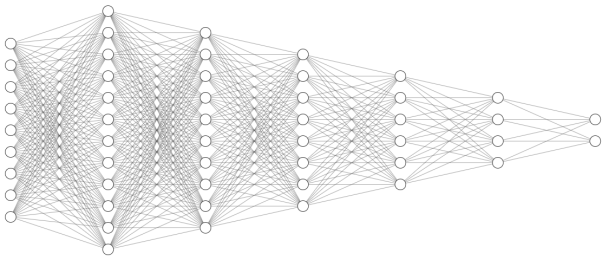


Figure 4: Representative neural network architecture [38]

Figure 4 illustrates the generic model architecture employed in this work. Models consisted of five hidden layers, [12, 10, 8, 6, 4], with two output neutrons in the final layer corresponding to reactor "on" and "off" states. The input layer was adjusted to evaluate model performance using different sets of input features. The baseline case included 9 input features corresponding to the 9 Merlyn sensing modalities—acceleration; ambient light; red, green, and blue color signals; magnetic field; pressure; proximity; and temperature—with the accelerometer and magnetometer signals characterized by the ℓ^2-norm of their components. The components of the magnetometer and accelerometer Merlyn data streams were alternatively ingested as 3 independent input features per modality along with the other sensing modalities (for a total of 13 input nodes) or as the sole input features (for a total of 6 nodes in the input layer).

The rectified linear unit was chosen as the activation function for the internal nodes because it avoids the vanishing gradient problem while providing quick convergence. A softmax activation function was used in the output layer to normalize unbounded real numbers into a probability distribution across the available classes.

4.4. Source Model Training and Hyperparameter Optimization

The source dataset consisted of HFIR Cycles 484 through 489, which covered the period from October 30, 2019 to September 11, 2020.[1] The dataset was partitioned into training and test sets using two different sampling procedures: stratified splitting and a chronological split. Stratified sampling was employed with 80% of the source dataset assigned as the training set and the remaining 20% as the test set. The training and test sets had a class prevalence approximately equivalent to that of the full dataset: 61% and 39% of samples corresponding to reactor "on" and "off" states, respectively. While this approach ensures randomization of training and test samples and guarantees similar class representation in training and test sets, the non-independent nature of the Merlyn time series data may introduce bias in test set performance evaluation for input data streams with high temporal autocorrelation. Figure 5 shows the temporal autocorrelation coefficient for a few select signals. Weather-related features such as temperature have high levels of temporal autocorrelation, whereas signals from the magnetometer and accelerometer are far less temporally autocorrelated.

The chronological splitting scheme avoids this source of bias, but at the risk of providing non-random and potentially non-representative training and test sets. For the chronological split, a two-week buffer was introduced to mitigate the impact of temporal auto-correlation on short time scales. That is, samples that are temporally proximate are likely to be similar due to the 16 Hz sampling rate of the Merlyn being far faster than the underlying state changes. Without a buffer, samples that are temporally proximate can appear in both the training and test sets. Since they are likely to be similar signals, the source model will have a much higher likelihood of generating the correct predictions on test set samples given that it was trained on similar or nearly identical samples. For the time split, the training set consisted of HFIR Cycles 484 to 488, and the test set consisted of HFIR Cycle 489. The class prevalence of the training and test sets were approximately equal: 49% and 51% of samples corresponding to reactor "on" and "off" states, respectively.

[1] At HFIR, a cycle is characterized by a run period of approximately 26 days followed by a refueling and maintenance outage.

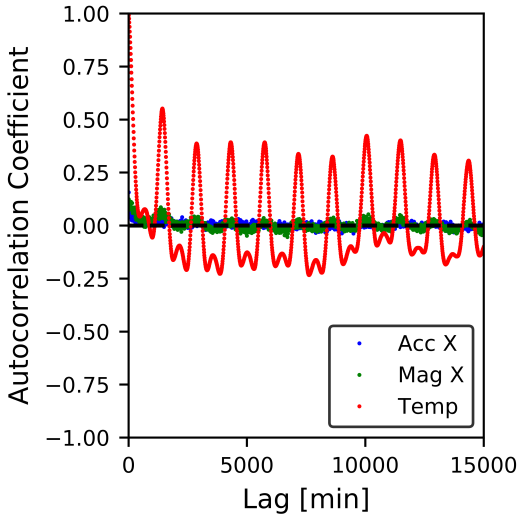


Figure 5: Autocorrelation coefficient as a function of lag in minutes for the accelerometer (x-component, blue), magnetometer (x-component, green), and temperature (red) signals after background correction has been applied.

Hyperparameter tuning was performed using an exhaustive grid search method, where networks were trained and evaluated on all possible permutations of candidate hyperparameters. Cross validation was performed using a stratified k-fold split. That is, the training data were separated into k=3 folds, each of which had roughly equal class prevalence. To ensure this, the training data were processed into batches and each batch was assigned a label corresponding to the majority class within the batch. These batches were then placed into the appropriate fold such that the prevalence of the batch classes roughly matched that of the original training set.

For a set of hyperparameters with possible values $c_1, \ldots, c_N$ and k number of k-fold splits, $k * c_1 * \ldots * c_N$ models were trained. Each model was trained until either 15 epochs of the training dataset had occurred or no significant improvement in predictions made on the validation set had been achieved for 5 epochs. The top-performing hyperparameter sets were selected by averaging cross-validation scores across each k-fold split and selecting the hyperparameter set with the highest score. The source model was constructed using optimized hyperparameters and then subsequently trained on the source training data. The hyperparameters tested during model generation are given in Table 2. The optimal hyperparameters for the source model

Table 2: Hyperparameter search space for the neural network classifier. Optimal hyperparameters are highlighted.

Hyperparameter	Value(s)
Training Batch Size	{1024, 4096, 8192, 32768} samples
Hidden Layers	{[12,10,8,6,4],[8,6,4],[8,6]}
Dropout Rate	{0.00, 0.10, 0.20}
Learning Rate	{0.0001, 0.001, 0.01}
Optimizer	Adam: $\beta_1 = 0.9$, $\beta_2 = 0.999$ [40]
Loss Function	Categorical Cross-Entropy

were determined as follows: a training batch size of 4096 samples, a dropout rate of 0.00, a learning rate of 0.001, and a hidden layer architecture of [12,10,8,6,4].

Parallelization was implemented in MIMOSAS for improved efficiency in hyperparameter tuning. A multiprocessing approach was selected over multi-threading as the grid search algorithm for hyperparameter estimation required training independent neural networks that do not share data. Multiprocessing enabled MIMOSAS to use multiple CPU cores to simultaneously train neural networks by copying over memory to each core and keeping the data confined to its respective core. For a common test script with six sets of hyperparameters, users experienced a 70% improvement in total training and evaluation time as compared to running the script serially. In general, the effectiveness of the parallelization implementation increased as the number of sets of hyperparameters increased.

4.5. Model Performance Evaluation

In nuclear proliferation detection, the absence of some derived signals is as important to identify as their presence. An accuracy metric for performance evaluation in statistical machine learning provides the fraction of correct classifications, but fails to properly characterize performance in classification problems with class imbalance. The Matthews Correlation Coefficient (MCC) was used as a performance metric in this work as it is useful in the presence of high disparity in class prevalence and penalizes both false positives and false negatives [41, 42]. An MCC of 1 indicates perfect performance, a value of -1 (in binary classification) results from exactly incorrect predictions, and a value of 0 corresponds to the classifier providing no improvement over randomly sampling the distribution of the class prevalences. As a performance metric, the MCC is interpreted similarly to a Pearson correlation coefficient [43].

5. Proxy Labeling Procedure

This section outlines the proxy labeling procedure that mediates transfer of the machine learning model. Proxy label generation is fundamental to the cluster-then-label approach to domain adaptation introduced in this work. Section 5.1 describes the feature selection analysis and modalities adopted for use in the feature mapping algorithm. The clustering algorithm is detailed in Section 5.2. Section 5.3 describes the derived features used as inputs to the clustering analysis. The cluster-then-label approach is detailed in Section 5.4. Finally, in Section 5.5, the transfer learning step is described and the effectiveness of the cluster-then-label approach for domain adaptation is investigated using synthetic labels.

5.1. Feature Selection

Dimensionality reduction was accomplished by down-selecting the most important input features as determined using Recursive Feature Addition (RFA) [44] and nuclear engineering considerations. Models were trained with identical internal architecture to the full model with each of the individual sensing modalities acting as the only input feature and the input layer size adjusted to the number of input features. The modality that most improved the MCC was selected for permanent inclusion. Models were then trained with two input features—the permanently-included feature and one candidate feature from the remaining input set—and the candidate input that provided the greatest improvement in the MCC over the single-input model was permanently added. Recursive implementation of this process provided a ranked importance of the input features.

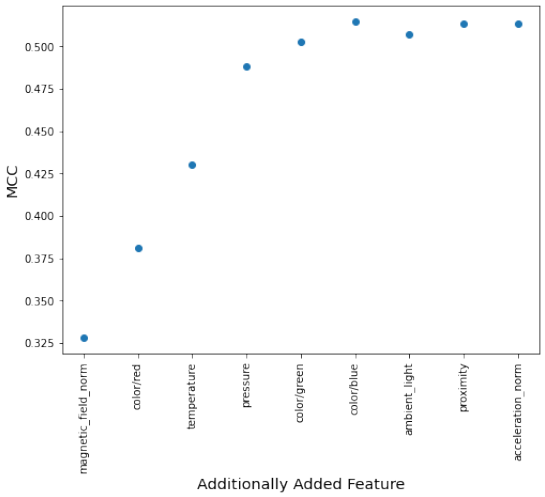


Figure 6: Matthews Correlation Coefficient as a function of the number of features included in the assessment using data obtained from HFIR-109 (Cycles 484-486). The ℓ^2-norm of the accelerometer and magnetometer signals were ingested as inputs.

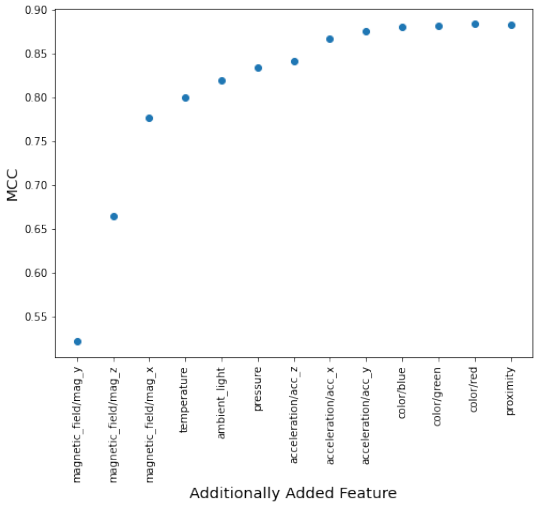


Figure 7: Matthews Correlation Coefficient as a function of the number of features included in the assessment using data obtained from HFIR-109 (Cycles 484-486). The vector components of the accelerometer and magnetometer signals were ingested as inputs.

The feature selection analysis was performed using a model trained and evaluated on a subset of the source data (i.e., HFIR Cycles 484-486). The RFA results are provided in Figures 6 and 7. For Figure 6, the analysis was performed using the ℓ^2-norm of the magnetic field and acceleration signals whereas the results in Figure 7 were generated by treating the vector components of the magnetometer and accelerometer signals as individual input features. In both analyses, the magnetometer signals were determined to be the most important input feature. Source model performance was higher when using the component input features (i.e., a maximum MCC of 0.88 as compared to 0.51 for vector magnitudes). Yet without alignment of the sensor axes between the testbed and target facilities, transfer of these input features does not have a strong physical basis. While other features, such as temperature, were consistently deemed important, results from the autocorrelation analysis suggest that their contribution to the classification performance may be derived in part from their temporal autocorrelation. As such, the ℓ^2-norm of the magnetic field signals was selected for cluster space feature mapping. In addition, the accelerometer signals were adopted for use in the clustering analysis as they are expected to correlate with vibrations from pump and fan motors associated with cooling tower operations, which is in turn correlated with reactor operations.

5.2. Clustering Algorithm

A variety of clustering algorithms were explored including k-means [45], Gaussian mixture models [46], spectral clustering [47], and density-based spatial clustering of application with noise (DBSCAN) [48]; among these, DBSCAN was adopted for use. DBSCAN operates by selecting an unclustered point from the dataset, assigning it a cluster identifier, and then assigning that same identifier to all points within a defined search distance, ϵ, from any point which has been assigned that identifier. Once no additional points can be added this way, the number of points with the identifier is compared to a user-defined minimum cluster population threshold and either enumerated as a cluster (if exceeding the threshold) or its constituent points are classed as "noise." One of the major advantages of DBSCAN is the lack of a requirement on cluster shape, which avoids the convexity requirement of other algorithms. The minimum cluster population was set to $2k-1$ using a heuristic from Sander et al. [49], where k is the dimension of the data. For the proxy-label cluster space, $k = 2$ since it is comprised of the ℓ^2-norm of the magnetic field and acceleration signals. The ϵ parameter was chosen by optimizing the silhouette score of the resulting clusters [50].

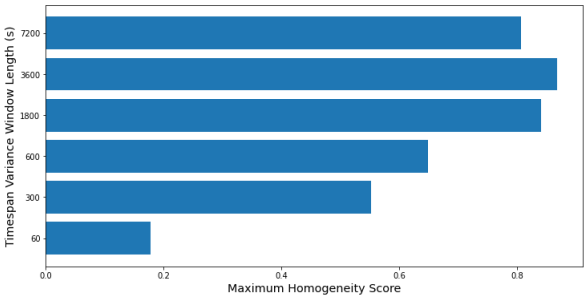


Figure 8: Time-series segmentation optimization for derived statistics used in the proxy-label generation cluster space. Scores represent the maximum homogeneity obtained by sweeping the DBSCAN search distance parameter, ϵ, from 0.001 to 0.1 in increments of 0.001.

To enable proxy labeling for the N classes within the classification problem (i.e., $N = 2$ for binary classification), the output of the DBSCAN algorithm was modified to consolidate the M clusters identified by DBSCAN into N distinct clusters that map directly to the N classes. That is, samples that did not belong to the N most populous clusters from the DBSCAN output were reassigned to their closest populous cluster centroid based on the Euclidean distance.

5.3. Temporal Statistics

To separate samples associated with "Reactor on" and "Reactor off" states, a derived statistic, $\vec{s}$, was defined based on the variance of the ℓ^2-norm of the magnetic field and accelerometer signals over a contiguous time segment:

$$\vec{s} = \begin{pmatrix} \mathrm{Var}\left(\|\vec{mag}\|_{t_1}^{t_2}\right) \\ \mathrm{Var}\left(\|\vec{acc}\|_{t_1}^{t_2}\right) \end{pmatrix}, t_1 \leq t_2. \tag{4}$$

To ensure a common range for both features in the cluster space, the variance data were scaled to the interval [0, 1] using min-max normalization. The duration of the time segment was investigated by clustering the variance-transformed data using the modified DBSCAN algorithm over 1-, 5-, 30-, 60- and 120-minute time segments. The cluster performance was then scored using class homogeneity as the evaluation metric [51] and the results are given in Figure 8. The highest homogeneity score was achieved using the 60-minute time segment.

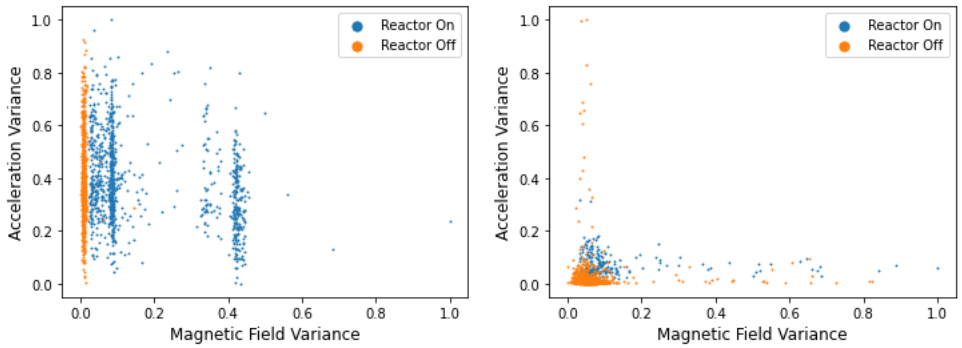
5.4. Cluster-then-label Method

The variance transformation (as described in Eq. 4) and min-max scaling were applied to the ℓ^2-norm of the magnetometer and accelerometer data from both the source and target datasets. For the resulting output of the modified DBSCAN algorithm, a cluster homogeneity score of 0.86 was achieved using the source dataset, and a clustering silhouette score of 0.48 was obtained for the target data. The source and target cluster spaces with samples labeled based on facility ground truth are given in Figure 9a. Clusters in both the source and target dataspaces were sorted in order of the distance of their centroids to the origin of each dataspace and then mapped across facilities based on their respective ranking in the sorted order. For the binary classification problem of reactor operational state, the cluster closest to the origin in the HFIR dataset was comprised in majority of data from the "Reactor Off" class whereas the more distant clusters consisted in majority of "Reactor On" samples.

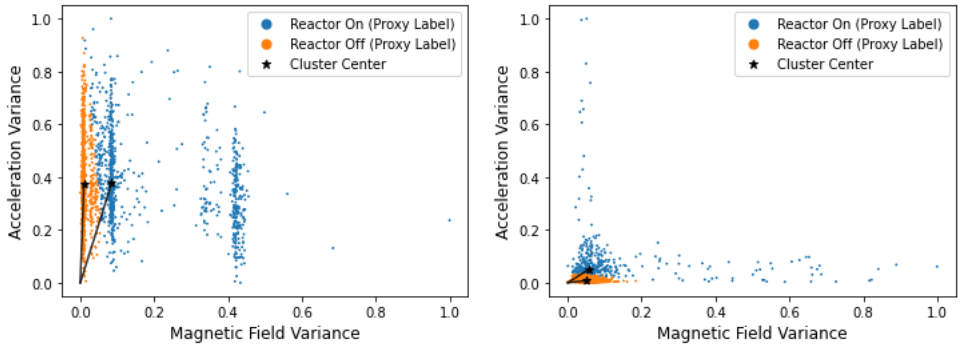
With cluster assignments between the source and target data in place, proxy-label assignments for the target samples were generated. Each target sample within a given target cluster was assigned a label based on the majority class within the corresponding source cluster. For example, a corresponding source cluster may consist of samples with a class prevalence of 80% "Reactor On" and 20% "Reactor Off"; in this case, the corresponding target cluster would be assigned the majority label: "Reactor On." A proxy-label accuracy of 88% was achieved when evaluated against the target ground truth. The results of the modified DBSCAN clustering and associated feature mapping are given in Figure 9b. These label values were then propagated back to each raw data point within the 60-minute time segment used to generate the derived dataspace.

5.5. Incremental Learning

The first ∼80% of the samples in the chronological target dataset were proxy-labeled and used as training data in the incremental learning step for 10 epochs. The test set consisted of the remaining 20% of the target dataset with the labels of the test set assigned based on the MNRC operator logs. A holdout buffer of two weeks was inserted between the training and test set to mitigate the impact of temporal autocorrelation. During the incremental learning step, the model parameters were allowed up update either with none of the original layer weights frozen or with only the first layer frozen to preserve source model weights. After in-



(a) HFIR-09 and MNRC-03 cluster space prior to application of density-based clustering. Samples are labeled with the ground truth and correspond to data within a 60-minute variance time segment.



(b) Modified DBSCAN clustering results with data labeled via the cluster-then-label approach. The black lines illustrate the distance from the origin to each cluster centroid.

Figure 9: Left: HFIR-09 source cluster space. Right: MNRC-03 target cluster space.

cremental training, the model was evaluated using the ground-truth labeled target test data.

The success of the incremental training step in adjusting model parameters to reflect the distinctive features of the target domain is dependent upon the ability to generate accurate proxy labels via the cluster-then-label procedure. To assess the dependence of target facility predictive success on the fidelity of the proxy labels, 11 copies of a testbed-trained model were incrementally trained on the target facility training dataset with labels varying in accuracy from 0% (labels are the exact opposite of ground truth) to 100% (labels match ground truth perfectly) in 10% increments. These label sets of varying fidelity were constructed by generating accurate labeled training data using ground truth from target facility operator logs, grouping these data into 60-minute segments (to approximate the structure of the proxy labeling scheme), and then inverting the labels of some fraction (e.g., 10%, 20%, etc.) of the samples on a per-segment basis. The incrementally-trained family of models was evaluated on the target facility test set, for which the labels were unadulterated.

The model used to assess the effect of proxy label accuracy had two hidden layers consisting of 8 and 6 nodes and used a batch size of 2^{15}, an initial learning rate of 0.001, and an exponential linear unit (ELU) activation function. A ten-fold cross-validation was performed for each synthetic dataset for uncertainty quantification. The HFIR model was also evaluated on the test set for each fold prior to incremental learning to provide a baseline transfer performance. Since the model consisted of two hidden layers, two tests were performed: one in which all weights and biases were allowed to update during incremental training, and another in which the first hidden layer of the model was frozen.

The results of the assessment of the effect of proxy label accuracy on transfer performance are shown in Figure 10. Both models (first- and no-layers frozen) showed improvement over baseline at and above 50% proxy label accuracy, with the largest marginal improvements in proxy-label per-

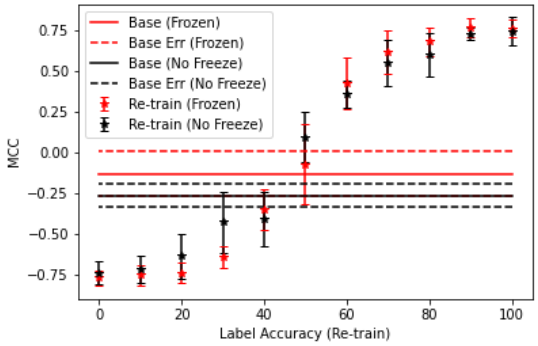


Figure 10: Effect of fidelity of labels generated using the "cluster-then-label" approach on the resulting model predictive performance.

formance between 60% and 80%. The model with the first layer frozen outperformed the model with no frozen layers, but this effect, though consistent as the accuracy of the synthetic proxy labels increased, was well within uncertainty. This increase in performance may indicate that freezing layers during incremental training increases the generalizability of the models. Finally, the performance evaluation using synthetic labels was compared against the model performance obtained using the cluster-then-label feature mapping algorithm. When no layers were frozen during training, the cluster-then-label approach resulted in an accuracy of 88% and an MCC of 0.71. These values agree within uncertainty with the synthetic label assessment, which predicted a MCC of 0.72 ± 0.04 at 90% label accuracy.

6. Results and Discussion

The source model performance was assessed, where the model was trained and evaluated using solely source data for both stratified and chronological splitting schemes, and the results are summarized in Table 3. In the baseline case (i.e., all input features with the accelerometer and magnetometer signals characterized by the ℓ^2-norm of their components), the stratified sampling yielded a MCC of 0.45 corresponding to a moderate correlation between the known and predicted labels. However, under the chronological split, the model performance notably dropped with a MCC of 0.25, representing a weak correlation between the known and predicted labels. Given that the temporal autocorrelation was relatively low in the magnetometer and accelerometer signals, a source model was trained and evaluated using the two different splitting schemes with only the accelerometer and magnetometers components as input features. In this case, the model performance was similar for both splitting schemes, suggesting that the classification performance obtained in the stratified sampling approach was not biased by temporally autocorrelated samples. Further, the model performance was strong, with a classification accuracy of $> 90\%$ and a MCC corresponding to a strong positive correlation between the known and predicted labels.

The results of the baseline transfer (i.e., direct application of the testbed-trained model to the target dataset) and the cluster-then-label approach to transfer learning are provided in Table 4 for various splitting schemes and input features. For the baseline case, (i.e., source models directly applied to the target dataset with no incremental learning), the classification performance was poor. Model accuracies were typically around 30% while the MCC ranged from -0.26 to 0.093. This is not unexpected as the information stored in the testbed-trained model does not reflect the differences in facility layout, equipment, and operations cycles between the testbed and target facilities. Transfer performance was similar across all adapted source models. This may suggest that the network weights of the source model are largely being overwritten by the incremental learning step. To evaluate this, a model was trained using only the proxy-labeled target data (i.e., no testbed-trained model and no incremental learning step) and then evaluated on the target test set. The resulting accuracy of 85% and MCC of 0.68 are consistent with the performance of the adapted model applied to the target dataset. To combat this "catastrophic forgetting," the learning rate may be adjusted in the incremental learning step such that the information from the source domain is not abruptly lost upon continued training [52]. Further, models trained incrementally with the first layer frozen performed similarly to those with all layers free, which indicates that the general features of the first layer of the source model contains features similar enough to the target domain so as not to impede classification performance. The proxy-labeled model as well as all adapted source models generally demonstrated strong classification performance, which highlights the success of the cluster-then-label approach for proxy label generation.

Split Scheme	Input Features	Accuracy	MCC
Stratified	All (ℓ^2-norm)	0.73	0.45
Chronological	All (ℓ^2-norm)	0.62	0.25
Stratified	Magnetic Field and Acceleration Components	0.91	0.81
Chronological	Magnetic Field and Acceleration Components	0.93	0.86

Table 3: Model trained and evaluated on source data under different sampling schemes for training and test set generation. Different input features were assessed to explore the impact of temporal autocorrelation.

Model	Split Scheme	Input Features	Accuracy	MCC
Source (Baseline)	Stratified	All (ℓ^2-norm)	0.32	-0.26
Adapted (No Layers Frozen)	Stratified	All (ℓ^2-norm)	0.84	0.69
Adapted (First Layer Frozen)	Stratified	All (ℓ^2-norm)	0.85	0.68
Source (Baseline)	Chronological	All (ℓ^2-norm)	0.30	-0.023
Adapted (No Layers Frozen)	Chronological	All (ℓ^2-norm)	0.83	0.65
Adapted (First Layer Frozen)	Chronological	All (ℓ^2-norm)	0.83	0.61
Source (Baseline)	Stratified	Mag/Acc Comp.	0.68	-0.096
Adapted (No Layers Frozen)	Stratified	Mag/Acc Comp.	0.86	0.69
Adapted (First Layer Frozen)	Stratified	Mag/Acc Comp.	0.86	0.69
Source (Baseline)	Chronological	Mag/Acc Comp.	0.30	0.093
Adapted (No Layers Frozen)	Chronological	Mag/Acc Comp.	0.85	0.68
Adapted (First Layer Frozen)	Chronological	Mag/Acc Comp.	0.85	0.69
Proxy-labeled Target Model		All (ℓ^2-norm)	0.85	0.68

Table 4: Source model transfer evaluation using training and test datasets obtained via stratified and chronological splitting. Baseline results were obtained through direct evaluation of the source model on ground-truth-labeled target data without any incremental learning or proxy-labeled target data. All (ℓ^2-norm) refers to use of all Merlyn sensor input features with ℓ^2-norm representations for magnetic field and acceleration. Mag/Acc Comp. indicates use of only the magnetic field and acceleration components as input features.

7. Summary

A transductive transfer learning technique was applied to characterize nuclear reactor operations. The impact of feed-forward neural networks trained under different splitting schemes and multisource input features were evaluated and compared. With stratified splitting, the operational status of the testbed reactor was determined with an accuracy and MCC of 0.73 and 0.45, respectively. Under chronological splitting, source model performance dropped to an accuracy and MCC of 0.62 and 0.25, respectively. After reducing the input feature space to magnetic field and acceleration components only with chronological splitting, the operational status of the reactor was determined with an accuracy and MCC of 0.93 and 0.86, respectively. Direct application of all source models to the target dataset resulted in poor classification performance, summarized in Table 4. For the transductive transfer, derived features were generated using the variance of the accelerometer and magnetometer measurements over 60-minute time segments. A modified density-based clustering algorithm was then applied to both the source and target data, and proxy labels were assigned using the known labels from the source domain, the cluster IDs of both the source and target data, and a cross-facility mapping created by ordering the cluster distances from the origin. All source models were further trained using incremental learning via ingestion of a subset of the proxy-labeled target data, and the classification accuracies and MCCs for the adapted models evaluated on the target test set were on average increased to 0.84 and 0.67, respectively. When the first layer of the source models were frozen, target domain classification performance was not significantly impacted. The accuracies and MCCs of all adapted models were similar to that of a proxy-labeled target model (trained only using the proxy-labeled target data), which achieved an accuracy and MCC of 0.85 and 0.68, respectively. The results obtained in this work outline a pathway for generalizing the applicability of multisource classification for a range of nuclear facilities and potential monitoring scenarios.

Acknowledgements

The authors thank the MINOS Venture team for their help in performing these experiments and recognize in particular Jared Johnson, Will Ray, and Michael Willis at Oak Ridge National Laboratory. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Berkeley National Laboratory under Contract DE-AC02-05CH11231. The project was funded by the U.S. Department of Energy, National Nuclear Security Administration, Office of Defense Nuclear Nonproliferation Research and Development (DNN R&D). This material is based upon work supported in part by the Department of Energy National Nuclear Security Administration through the Nuclear Science and Security Consortium under Award Number DE-NA0003180. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

[1] United Nations Office for Disarmament Affairs, Treaty on the Non-Proliferation of Nuclear Weapons, https://www.un.org/disarmament/wmd/nuclear/npt (1968).

[2] L. Rockwood, Legal Framework for IAEA Safeguards, International Atomic Energy Agency, Vienna, 2013.
URL https://www.iaea.org/publications/10388/legal-framework-for-iaea-safeguards

[3] International Atomic Energy Agency, Nuclear Forensics in Support of Investigations, no. 2-G (Rev. 1) in Implementing Guides, Vienna, 2015.
URL https://www.iaea.org/publications/10797/nuclear-forensics-in-support-of-investigations

[4] V. Fedchenko, The role of nuclear forensics in nuclear security, Strategic Analysis 38 (2) (2014) 230–247. doi:10.1080/09700161.2014.884442.

[5] C. L. Stewart, B. L. Goldblum, Y. A. Tsai, S. Chockkalingam, S. Padhy, A. Wright, Multimodal data analytics for nuclear facility monitoring, in: Proceedings of the Institute of Nuclear Materials Management 60th Annual Meeting, Palm Springs, INMM, 2019, pp. 1024–1034.

[6] S. B. Kotsiantis, Supervised machine learning: A review of classification techniques, in: Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies, IOS Press, NLD, 2007, pp. 3–24.

[7] J. Huang, Q. Dong, S. Gong, X. Zhu, Unsupervised deep learning by neighbourhood discovery, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, Vol. 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 2849–2858.
URL http://proceedings.mlr.press/v97/huang19b.html

[8] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (10) (2010) 1345–1359. doi:10.1109/TKDE.2009.191.

[9] R. Vilalta, C. Giraud-Carrier, P. Brazdil, C. Soares, Inductive Transfer, Springer US, Boston, MA, 2010, pp. 545–548. doi:10.1007/978-0-387-30164-8_401.
URL https://doi.org/10.1007/978-0-387-30164-8_401

[10] A. Arnold, R. Nallapati, W. W. Cohen, A comparative study of methods for transductive transfer learning, in: Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), 2007, pp. 77–82. doi:10.1109/ICDMW.2007.109.

[11] N. FarajiDavar, T. de Campos, J. Kittler, F. Yan, Transductive transfer learning for action recognition in tennis games, in: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011, pp. 1548–1553. doi:10.1109/ICCVW.2011.6130434.

[12] M. Rohrbach, S. Ebert, B. Schiele, Transfer learning in a transductive setting, in: C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 26, Curran Associates, Inc., 2013, pp. 1–9.
URL https://proceedings.neurips.cc/paper/2013/file/3295c76acbf4caaed33c36b1b5fc2cb1-Paper.pdf

[13] N. Farajidavar, Transductive transfer learning for computer vision, Ph.D. thesis, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom (7 2015).

[14] G. Coley, BeagleBone Black System Reference Manual, Beagleboard, available at https://cdn.sparkfun.com/datasheets/Dev/Beagle/BBB_SRM_C.pdf, Rev. C.1 (1 2014).

[15] Atmel Corporation, 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash, available at https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf, Rev. 7810D-AVR-01/15 (1 2015).

[16] ROHM Semiconductor, SensorShield-EVK-003 Manual, available at http://rohmfs.rohm.com/en/products/databook/applinote/ic/sensor/sensorshield-evk-003_ug-e.pdf, Rev.001 (4 2018).

[17] Kionix, 8g / 16g / 32g Tri-axis Digital Accelerometer Specifications, available at https://d10bqar0tuhard.cloudfront.net/en/datasheet/KX224-1053-Specifications-Rev-2.0.pdf, Rev. 2.0 (12 2017).

[18] ROHM Semiconductor, Optical Proximity Sensor and Ambient Light Sensor with IrLED, available at https://fscdn.rohm.com/en/products/databook/datasheet/opto/optical_sensor/opto_module/rpr-0521rs-e.pdf, Rev.001 (1 2016).

[19] ROHM Semiconductor, Ambient Light Sensor IC Series: Digital 16bit Serial Output Type Color Sensor IC, available at https://fscdn.rohm.com/en/products/databook/datasheet/ic/sensor/light/bh1749nuc-e.pdf, Rev.002 (12 2017).

[20] Adafruit, Adafruit Ultimate GPS Featherwing, available at https://cdn-learn.adafruit.com/downloads/pdf/adafruit-ultimate-gps-featherwing.pdf (6 2020).

[21] ROHM Semiconductor, Magnetic Sensor Series: 3-Axis Digital Magnetometer IC, available at https://fscdn.rohm.com/en/products/databook/datasheet/ic/sensor/geomagnetic/bm1422agmv-e.pdf, Rev.001 (10 2016).

[22] ROHM Semiconductor, Pressure Sensor series: Pressure Sensor IC, available at https://fscdn.rohm.com/en/products/databook/datasheet/ic/sensor/pressure/bm1383aglv-e.pdf, Rev.003 (3 2016).

[23] ROHM Semiconductor, Temperature Sensor IC, available at https://fscdn.rohm.com/en/products/databook/datasheet/ic/sensor/temperature/bd1020hfv-e.pdf, Rev.001 (11 2015).

[24] J. Hite, K. J. Dayman, N. S. Rao, C. Greulich, S. Sen, A. D. Nicholson, D. E. Archer, J. Johnson, D. Chichester, M. J. Willis, I. Garishvili, A. Rowe, J. Ghawaly, Automated vehicle detection in a nuclear facility using low-frequency acoustic sensors, Tech. rep., Oak Ridge National Laboratory (7 2020).
URL https://www.osti.gov/biblio/1649300

[25] Multiple Informatics for Nuclear Operations Scenarios Database, https://minos.lbl.gov.

[26] Berkeley Data Cloud, https://bdc.lbl.gov/.

[27] E. Covas, Transfer learning in spatial–temporal forecasting of the solar magnetic field, Astronomische Nachrichten 341 (4) (2020) 384–394. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/asna.202013690, doi:https://doi.org/10.1002/asna.202013690.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/asna.202013690

[28] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, JMLR.org, 2015, p. 1180–1189.

[29] M. Peikari, S. Salama, S. Nofech-Mozes, A. L. Martel, A cluster-then-label semi-supervised learning approach for pathology image classification, Scientific Reports 8 (1) (2018) 7193. doi:10.1038/s41598-018-24876-0.

[30] Z. Li, D. Hoiem, Learning without forgetting, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (12) (2018) 2935–2947. doi:10.1109/TPAMI.2017.2773081.

[31] J. Zhao, C. Stewart, B. Goldblum, A. Y. Tsai, S. Chockkalingam, P. V. Valdez, MIMOSAS (Sept. 2019).
URL https://github.com/nonproliferation/mimosas

[32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL https://www.tensorflow.org/

[33] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, Array programming with NumPy, Nature 585 (2020) 357–362.

doi:10.1038/s41586-020-2649-2.

[34] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. doi:10.1038/s41592-019-0686-2.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[36] A. Collette, Python and HDF5, O'Reilly Media, Inc., 2013.
URL https://www.h5py.org/

[37] Climate Data Online, https://www.ncdc.noaa.gov/cdo-web/ (5 2017).

[38] A. LeNail, Nn-svg: Publication-ready neural network architecture schematics, Journal of Open Source Software 4 (33) (2019) 747. doi:10.21105/joss.00747.
URL https://doi.org/10.21105/joss.00747

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15 (1) (2014) 1929–1958.

[40] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, arXiv e-prints (2014) arXiv:1412.6980arXiv:1412.6980.

[41] B. Matthews, Comparison of the predicted and observed secondary structure of t4 phage lysozyme, Biochimica et Biophysica Acta (BBA) - Protein Structure 405 (2) (1975) 442 – 451. doi:https://doi.org/10.1016/0005-2795(75)90109-9.
URL http://www.sciencedirect.com/science/article/pii/0005279575901099

[42] S. Boughorbel, F. Jarray, M. El-Anbari, Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric, PLOS ONE 12 (6) (2017) 1–17. doi:10.1371/journal.pone.0177678.
URL https://doi.org/10.1371/journal.pone.0177678

[43] D. M. W. Powers, Evaluation: From precision, recall and F-factor to ROC, informedness, markedness & correlation, Journal of Machine Learning Technologies 2 (2011) 37–63.

[44] T. Hamed, Recursive Feature Addition: a Novel Feature Selection Technique, Including a Proof of Concept in Network Security, Doctoral dissertation, The University of Guelph (2017).

[45] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, Society for Industrial and Applied Mathematics, USA, 2007, pp. 1027–1035.

[46] D. A. Reynolds, T. F. Quatieri, R. B. Dunn, Speaker verification using adapted Gaussian mixture models, Digital Signal Processing 10 (1) (2000) 19–41. doi:https://doi.org/10.1006/dspr.1999.0361.
URL https://www.sciencedirect.com/science/article/pii/S1051200499903615

[47] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, NIPS'01, MIT Press, Cambridge, MA, USA, 2001, pp. 849–856.

[48] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.

[49] J. Sander, M. Ester, H.-P. Kriegel, X. Xu, Density-based clustering in spatial databases: The algorithm gdbscan and its applications, Data Mining and Knowledge Discovery 2 (2) (1998) 169–194. doi:10.1023/A:1009745219419.

[50] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65. doi:https://doi.org/10.1016/0377-0427(87)90125-7.
URL https://www.sciencedirect.com/science/article/pii/0377042787901257

[51] A. Rosenberg, J. Hirschberg, V-measure: A conditional entropy-based external cluster evaluation measure, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 410–420.
URL https://www.aclweb.org/anthology/D07-1043

[52] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell, Overcoming catastrophic forgetting in neural networks, Proceedings of the National Academy of Sciences 114 (13) (2017) 3521–3526. arXiv:https://www.pnas.org/content/114/13/3521.full.pdf, doi:10.1073/pnas.1611835114.
URL https://www.pnas.org/content/114/13/3521