3-2021

Application of Artificial Neural Networks to Elemental Assay Data for Nuclear Forensics Analysis

Jason G. Seik

Application of Artificial Neural Networks to Elemental Assay Data for Nuclear Forensics Analysis

THESIS

Jason G. Seik, Captain, USAF

AFIT-ENP-MS-21-M-135

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENP-MS-21-M-135

APPLICATION OF ARTIFICIAL NEURAL NETWORKS TO ELEMENTAL ASSAY DATA FOR NUCLEAR FORENSICS ANALYSIS

THESIS

Presented to the Faculty

Department of Engineering Physics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Nuclear Engineering

Jason G. Seik, B.S.

Captain, USAF

March 2021

AFIT-ENP-MS-21-M-135

APPLICATION OF ARTIFICIAL NEURAL NETWORKS TO ELEMENTAL ASSAY DATA FOR NUCLEAR FORENSICS ANALYSIS

THESIS

Jason G. Seik, B.S.
Captain, USAF

Committee Membership:

Abigail Bickley, Ph.D.
Chair

Brett Borghetti, Ph.D.
Member

Ashley Holland, Ph.D.
Member

John McClory, Ph.D.
Member

AFIT-ENP-MS-21-M-135

Abstract

Artificial neural networks are applied to elemental assay data of microscopic, actinide bearing particles obtained using energy dispersive x-ray spectroscopy via a scanning electron microscope (SEM-EDS) and Electron Probe Micro Analysis (EPMA). This technique provides a non-destructive assessment of the composition of particles that is suitable for nuclear forensics applications because it preserves the high-value material for subsequent feature extraction. This produces large volumes of data that are difficult for human analysts to ingest on the timescale required by national-level decision and policy makers. Consequently, it is advantageous to assess the applicability of advanced computational techniques to this large, numerical dataset.

An Artificial Neural Network (ANN) was used to compare and group like-particles together using a Siamese network and triplet loss function. The results were then compared to more traditional machine learning techniques, including logistic regression, linear discriminate analysis, decision tree, random forest, extra forest, and voting classifiers.

Each sample, or collection of particles of similar origin, is labeled with a serial code. The measure of success of each technique was the accuracy with which the model could predict the serial code, or group like-particles together based purely on the elemental assay of each particle observation. Using moment transformation provided up to a 7 times increase in accuracy. Models using a binary cross-entropy loss function showed higher accuracy when compared to a triplet loss function, but were very dependent on the type of data used for training. Training on data collected via EPMA gave accuracies 6 times greater than random chance, and training on data collected via SEM-EDS gave accuracies up to 60 times greater than random chance.

Dedicated to my wife and daughter.

Acknowledgements

I would like to thank my wife and daughter for their continued support and understanding while I spent many days, nights, and weekends working on research.

I would like to thank my advisor and thesis committee for keeping me on track through all of my work.

Finally, I would like to thank AFTAC and McCrone Labs for providing me the data for this research, and AFTAC for sponsoring this research.

Jason G. Seik

Table of Contents

Page

Page

List of Figures

List of Tables

APPLICATION OF ARTIFICIAL NEURAL NETWORKS TO ELEMENTAL ASSAY DATA FOR NUCLEAR FORENSICS ANALYSIS

I. Introduction

1.1 Overview

There is a constant concern with countries and terrorist organizations obtaining nuclear fuel for use in weapons. With nuclear power becoming more common it is important to understand where fissile material could have come from and what is implied by its presence. Particles of material obtained from a nuclear plant that generates power can be visually similar to those obtained from a plutonium generation facility, but utilizing the unique elemental signatures that each particle possesses can help determine its origin. This is one of the basic concepts of nuclear forensics.

Elements and their respective isotopes are found in different relative abundances across the world [1]. A soil sample from California, USA will have a different composition compared to a soil sample from Great Britain, UK, which will have a different composition compared to a soil sample from Hokkaido, Japan. These soil concentrations are, in essence, fingerprints for the location from which they originate [2]. Further, these concentrations are not just an indication of geographical origin but also of industrial or agricultural impacts caused by human activities. The concept of human impacts on the environment is referred to as anthropogenic. These anthropogenic impacts do not have to be large in order to be detectable. Trace elements can be indicators of nearby chemical or nuclear processing plants [3]. It is these fingerprints which enable nuclear forensics. Theoretically, if every area of the world has

a unique elemental assay then any unknown sample can be statistically correlated to a geographical point of origin, and if every material processing method produces a unique elemental assay, then any unknown sample can be traced back to the type of process that generated it.

For this research, each sample consists of a number of particles collected at the same time from the same geographical location. The number of particle observations is determined by how the sample is loaded into the measurement equipment and the technique used to identify and locate particles. An automated Scanning Electron Microscope coupled with Energy Dispersive X-Ray Spectroscopy (SEM-EDS) can extract tens of thousands of observations from a single sample, but many particle observations may not contain the elements of interest. On the other hand, a technician can isolate a particle of interest and measure its elemental composition using an Electron Probe Micro Analysis (EPMA). EPMA produces observations more representative of particles of interest, but typically yields less than a hundred particle observations. Each system measures a number of elemental features based on user input, which can lead to more than 30 features for a single particle observation. All of this can yield a very large amount of information, and it is not always realistic for a human to study the data and recognize a meaningful pattern within a reasonable amount of time. However, advanced machine learning techniques could be used to analyze the data and find patterns much more easily, and in less time, than a human.

1.2 Motivation & Problem Statement

The various isotopes of uranium and plutonium are important factors when trying to characterize a nuclear processing plant, but these alone are not enough to create a unique signature. Uranium and plutonium will decay into other actinides and spontaneously fission into lighter elements. For example, the uranium isotope U-238

follows a decay chain which results in 17 different isotopes, all of which can be used as a ^{238}U signature.

There are also non-fuel elements that are important in the fabrication of nuclear fuel. Combining the fuel and non-fuel factors can result in a comprehensive characterization of a nuclear plant and its intended purposes. However, an important issue is that these characterizations often need to be done on foreign nuclear plants and within the restraints allowed by nuclear treaty monitoring protocols. Because of this, there are a number of considerations that need to be addressed. First, there is never a guarantee that a large number of samples will be collected. Second, environmental contaminants are often intermingled with the actual sample of interest. Third, sample analysis must often be conducted in the blind without precise knowledge of the sample origins.

The first consideration cannot be controlled for. Nuclear treaty monitoring protocols do not always allow for the collection of large numbers of samples. To circumvent this issue, nuclear analysis teams try to extract as much meaningful data as possible from the available samples, and it is traditionally assumed that the isotopic information will give the most useful data. The issue with this lies in how the information is obtained.

For mass spectrometry techniques, there needs to be an isotope of interest that an analyst is specifically looking for. First, chemical separation may need to be performed to isolate a single, or few, elements. Whether or not chemical separation is performed, the sample is then ionized by some means and sent through high powered magnets to separate the isotopes based on the charge to mass ratio of the nucleus. Depending on the size of the collected sample, it can be split and this process can be done multiple times, but this process destroys the sample and does not produce a high number of observations per sample.

An alternative is to bombard the sample with photons or electrons. This results in emitted x-rays, which can be measured to determine the elemental composition based on energies or wavelengths. This can be done to each particle in the sample to obtain an observation. The method is non-destructive and can be done with much larger samples, but only yields elemental data. This elemental data is useful but does not describe the quality of nuclear fuel, which requires isotopic abundances to determine ^{235}U enrichment level.

To summarize the first consideration, current processes either produce too few observations from a single sample or do not produce enough fidelity on isotopic information to be useful with current nuclear forensic techniques. It is assumed, however, that a meaningful pattern using the low fidelity elemental data can sufficiently distinguish a unique nuclear fingerprint. A goal of this research is to either prove or disprove this assumption through the use of advanced machine learning techniques.

The second consideration involves how to deal with elemental noise present in the samples. The issue here is that without knowing sample origins there is no way to determine which elements present are noise and which are characteristic of the sample of interest. Even though local soil compositions have been characterized for many parts of the world, localized variation can be present [1]. First a database of soil elemental assays would need to be established that is easily compatible with current nuclear forensics databases. There would need to be a way to accurately determine the type of soil found in the collected sample. Then a method would need to be developed for appropriately removing soil signatures that account for statistical variations. Though this is an area of interest, this would diverge greatly from the intent of this research. It is important to keep in mind that samples will very likely have some amount of noise, but it is assumed that any signature within a sample is sufficiently distinguishable from the noise that a machine learning program can still

accurately characterize the sample.

The third consideration, identification of sample creation process, is considered next. For nuclear forensics, this means those physical and chemical processes that actually created the sample; what fuel fabrication process was used, what type of fuel was employed by the power plant, how the ore was mined, etc. The issue with this identification is that when a nuclear sample is received the true origins of the sample are usually unknown so the results can not be compared directly to the truth. Over the years, robust databases have been established linking sample data to sample origins, but this does not directly help in the identification of a new unknown sample.

A team of experts could characterize each new sample, but this is time consuming and may not be feasible within the time limits set by national-level decision and policy makers. An alternative to this could be to use machine learning to quickly compare this new unknown with the current database to determine if it is the same or similar to anything that has already been characterized.

1.3 Description of the Data & Machine Learning Approach

The data provided for this analysis is labelled with a serial code. This is simply used to distinguish one sample from the next. From each sample, either an automated SEM-EDS or EPMA can extract many different observations, each of an individual particle. Each observation is the particle's elemental composition and is essentially a smaller portion of the total sample. Each observation is assumed to be distinct and independent from one another, and the cumulative set of observations from a given sample is the distinct sample fingerprint. Figure 1 gives a visual of terminology relevant to this project. From here out, the terms sample and class will be used synonymously with a single sample that contains independent observations of many individual particles.

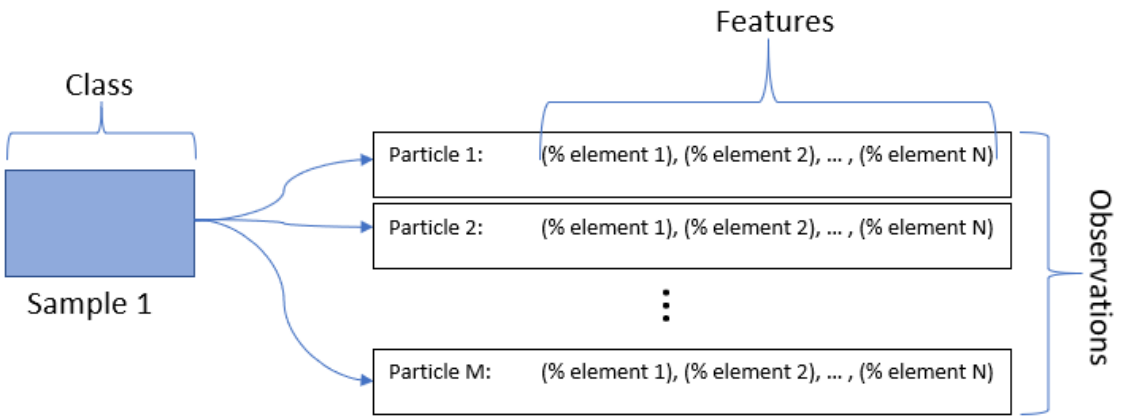

Figure 1. Each class represents a sample and is broken into M different particle observations. Each observation represents just a single piece of the whole sample and is characterized by N different elemental features. Each feature represents the percent composition of a given element present in a single observation.

Each observation consists of the elemental assay data. This means that each observation has a number of features equal to the total number of elements measured by the SEM-EDS or EPMA. Each feature is labeled with a single, specific element and consists of a single value between zero and one. This number represents the relative amount of the element found within the observation. This means that the sum of all features for a single observation will be 1.0, or 100 percent.

The basic way a machine learning program works is that it takes a set of data and makes a prediction on what that data most likely represents. In order to achieve this, the program needs to be trained on a set of data where the output is already known. It essentially creates a function based on the relationships observed between all the inputs (x values) and all outputs (y values). In regards to the nuclear forensics data, the input will consist of a matrix, with each row representing a particle observation and each column representing a feature, or in this case element from the periodic table.

In an ideal situation, the outputs would be the true sample origins, but that is

unavailable for this data. Instead, there are methods which compare the unknown data to known data. A common method for tackling this type of problem is to use a type of Artificial Neural Network (ANN) called a Siamese Network. This type of network takes the observations from one sample and compares them to the observations from another sample. If the observations are similar, then the Siamese Network will output 'same.' If the observations are not similar, then the Siamese Network will output 'not-same.' This effectively circumvents the issue of not having true sample origins.

Siamese Networks can be very effective, but often require each input to be representative of the entire sample. A single particle observation from a sample would not suffice. In order to circumvent this issue, the set of all particle observations from a single sample can be transformed into a more manageable meta observation using a set invariant transform. This set can then be fed into the Siamese Network.

1.4 Research Hypothesis

Everything stated thus far describes a current issue with evaluating elemental assays to determine nuclear processes and a potential solution using ANNs. Put succinctly:
As an initial proof of concept, an algorithm can be designed to distinguish how an uncharacterized sample of unknown origins compares to a well-defined and characterized sample with origins from some stage of the nuclear fuel cycle. Advanced statistical machine learning techniques and neural networks will be used to analyze elemental assays as input features (x values) and same/not-same probability as the outputs (y values).

II. Theory

2.1 Background

Much work has been done in the field of nuclear forensics and how to best attribute nuclear products to their point of creation within the nuclear fuel cycle, though the work is very time consuming and can take a dedicated team years to complete the analysis [4, 5]. Even so, this process does not always give definitive results on sample origin. It is difficult to attribute a location of origin to a sample based on common elements, such as carbon or oxygen. It has been shown that the rare earth elements provide the best means of unique characterization [6]. In order to acquire this information from a collected sample, instruments such as Scanning Electron Microscopes (SEMs) are used.

Various machine learning techniques can be used to decipher patterns within data. There are machine learning tools that can make a simple prediction based on given input values. For instance, a class can be predicted based on an elemental assay input. There exists a wide selection of basic machine learning techniques, including linear and polynomial models, decision tree classifiers, and culmination classifiers, each with their own strengths and weaknesses.

When these relatively simple models are not enough to make accurate predictions, or when the problem involves more than just predicting the class to which a given set of features belong, ANNs can be used. The idea behind ANNs is that they mimic the basic concepts of how a brain work; many individual nodes that do not work well on their own, but when brought together can make advanced predictions and correlations, or even serve as a means of generating artificial data [7].

2.2 Data Collection Instruments

To acquire elemental assay data of microscopic particles, a popular, practical, and convenient method is to use an SEM-EDS or EPMA; there are a number of instruments capable of collecting elemental assay data, but SEM-EDS and EPMA will be the focus of this work due to the achievable resolution of the devices. In an SEM the portion of the sample being analyzed is bombarded with a beam of monoenergetic electrons. These electrons scatter either elastically or inelastically. Elastic scattering will result in electrons with comparable energy to the incident electron. These are called backscatter electrons and typically result from direct collisions with the target nuclei. Inelastic scatter will result in electrons with much lower energies. In inelastic scattering, incident electrons can impart some of their energy to electrons within the atomic electron clouds of the target material, creating secondary electrons that can be measured. The energy of the secondary electrons will be directly dependent on the energy level of the electron shell they are ejected from and the energy of the incident electron. Detectors are set up around the target to measure the scattering angle and energies of these electrons. Based on the scattering angle and final energy, a calculation can be performed to recreate the sub-microscopic structure of the sample's surface. When electrons are ejected from the electron cloud, additional electrons will move to fill in the gaps. This causes x-rays to be emitted. The number and energy of these x-rays is dependent on which elements are present in the sample and is the source of the term Energy Dispersive X-Ray Spectroscopy (EDS) [8].

Some of the data analyzed in this study were collected using a TESCAN Mira 3 SEM-EDS that can produce anywhere from a few hundred to tens of thousands of particle observations from a single sample on the order of minutes. The exact amount of time is dependent on how many particles are present and the measurement residence time at each grid location. Each observation is characterized by a number

of different elemental assay[1] features equal to the number of elements measured by the SEM-EDS. The TESCAN Mira 3 can scan high resolution images down to the submicron range. At optimal settings, a 1.2 nm resolution can be obtained with a 30 keV electron beam, or a 1.5 nm resolution with a 1 keV electron beam. The pressure of the sample chamber can be adjusted to assist in dealing with a wide range of sample properties, such as outgassing or highly conductive material [9].

Another method for collecting similar assay data is EPMA. This method uses the same initial principles as the SEM-EDS, except instead of using energies to determine elemental assay, it uses the wavelength of the resulting x-rays, or Wavelength Dispersive X-Ray Spectroscopy (WDS). This method takes advantage of Bragg's Law, which essentially states that if an x-ray with a wavelength comparable to atomic spacing is incident on a crystalline lattice, then at a certain angle the diffracted x-rays will interfere constructively to produce a more intense peak. A number of different diffraction materials can then be set up around the sample with detectors positioned to capture the diffracted x-rays. Noting which detectors received a signal, one can determine the wavelength of the x-ray using the equation associated with Bragg's Law:

$$n\lambda = 2dsin\theta, \tag{1}$$

where θ is the glancing angle, d is the lattice spacing of the diffractive material, and $n\lambda$ is an integer multiple of the x-ray wavelength. The wavelengths can then be directly associated with the elements that produced the x-rays [10]. However, this does require more effort when trying to obtain data on a large number of elements. Often, many different WDSs are set up in an EPMA to ensure most elements can be

[1]Each elemental assay is the percent composition of every element found within the observation with one exception: an amount of oxygen is assumed and added on top of what is already found in each observation.

analyzed, and an EDS is used to capture the rest of the elements.

The EPMA used to acquire data for this project is a JEOL JXA-8200 with five WDSs and one EDS. WDS gives a much higher resolution over EDS, so elements that produce similar wavelength x-rays can be distinguished and trace elements can be detected with higher resolution. Like the TESCAN Mira 3, each observation is characterized by a number of different elemental assay features equal to the number of elements measured by the device. The biggest difference comes in how many observations are made. The TESCAN will take many different observations of the material placed into the machine, and can typically handle the entire sample at once. Conversly, the JXA-8200 essentially does a survey of the material placed in the machine, but cannot handle the same volume of material as the TESCAN. Because of this, samples are pre-processed to isolate particles of interest before they are placed into the JXA-8200. Each isolated particle is separately analyzed by the machine to produce a relatively small number of observations. However, because of the time and effort involved in the analysis, it is not typical for the entire sample to be analyzed. So the isolated particles are the ones that show signs of elements of interest. This process results in observations that are more representative of the material placed within the machine, but not necessarily statistically representative of the whole sample. The JXA-8200 can generate image resolutions as low as 5nm using the EDS, and can detect x-ray wavelengths from 0.087nm to 9.3nm using the WDS. Compared to the TESCAN, this provides a wider range of elements that can be reliably detected, such as elements as light as beryllium [11].

2.3 Basic Machine Learning Techniques

In order to solve complex classification problems, a number of techniques have been developed. The open source programming language, Python, has a package

called Scikit-Learn, which provides these techniques in an easy to use fashion [12]. The following Scikit-Learn training models were used for this research: logistic regression, linear discriminate analysis, decision tree classifier, random forest classifier, extra forest classifier, and voting classifier.

2.3.1 Linear Models

Linear models are ideal when making correlations in simple datasets. Because of their prevalence, they are easily available and easy to implement. For example, how far an object has traveled based on how long it has traveled (assuming a relatively consistent speed) is a linear correlation that can easily be tested. However, many real-world datasets do not have a simple linear correlation. For instance, traveling objects will likely accelerate and decelerate frequently throughout their journey, which is far from a linear correlation. The further from linearity a dataset is, the worse the accuracy will be of a linear model [13]. Two linear models were used to set a baseline for this work: logistic regression and linear discriminate analysis.

Logistic regression is a method that can take normalized feature inputs and gives a number between zero and one that represents the probability that the given features correspond to a specific class [13]. The basic concept of logistic regression is represented by the equation:

$$p(X) = \frac{e^{\beta X}}{1 + e^{\beta X}}, \tag{2}$$

where β contains the unknown model parameters, X is the input data, and $\hat{p}(X)$ is predicted output. If X is an array with each entry corresponding to a different feature, then β will be a vector of the same length also with each entry corresponding to a different feature. There can also be z number of X inputs, which requires z number of corresponding β values. This highlights the simple case of having only two categories to choose from. However, this concept can be extended to include any

number of classes.

The β values are estimated using a form of gradient descent. There are a number of options available through Scikit-Learn. The Limited-Memory Broyden–Fletcher–Goldfarb–Shanno Bound-Constrained (L-BFGS-B) algorithm was used for this work. L-BFGS-B is a popular and effective algorithm in the field of machine learning for its ability to support l2 normalization and effectiveness with large bound-constrained datasets [14, 15]. L2 normalization is the root sum square of a set of numbers, and it is sometimes called the Euclidean norm. It is easy to calculate the derivatives of the l2 norm, which makes it very useful for gradient based learning. A bound-constrained dataset is one that has distinct numerical cutoffs for its data. For instance, the data for this work will never be more than 100% or less than 0%.

Logistic regression does have limitations and can be unstable in instances when each class is too distinct or if inputs are distributed normally. 'Too distinct' in this situation refers to classes that have no overlap with one another. For example, comparing the characteristics of a human child to the characteristics of green algae, there will be very little to no overlap of characteristics. This makes linear discriminate analysis a popular counter to logistic regression since this model is more stable in these instances [13]. There is uncertainty in how the data for this work is distributed and how distinct each sample is, so having both of these linear models ensures that a wide range of samples can be accommodated.

Linear discriminate analysis makes the assumption that the inputs for each class are normally distributed and decision boundaries exist where the probabilities of any pair of differently-classed Gaussian distributions is equal. This is an approximation of what is known as a Bayes decision boundary, or Bayes classifier. This implies that each class has its own unique multivariate Gaussian function. Scikit-Learn employs a few ways to solve for the mean and standard deviation, but for this work eigenvalue

decomposition was the chosen method because it can handle classification problems and supervised dimensionality reduction. This creates a reproducible solution since no random seed is required for solving these matrix equations [12, 13]. Most other machine learning applications require some form of random number generator, known as a random seed, to initiate its algorithm.

2.3.2 Decision Trees & Ensemble Classifiers

When it comes to classification problems, a decision tree model can be a powerful tool. The basic idea is that there are nodes in the decision tree that check whether the input features meet specific criteria. These nodes will branch out and additional checks will be made, and each path down the decision tree corresponds to a different class. Figure 2 shows a simple example of how a decision tree works.

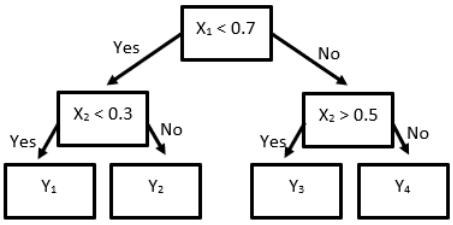

Figure 2. Two layers in a decision tree can separate data into four classes.

In order for an algorithm to build a decision tree, it first looks at all of the data and splits it into two regions with the goal of minimizing the error in each region. For example, this first split can be of the uranium composition. Region one could contain samples with a uranium composition greater than 70%, and region two could contain samples with uranium less than 70%. Each separate region is then split again in the same fashion. The split could be made using the same elemental feature as before, a different elemental feature altogether, or a combination of many different features. This process continues until each class is represented by its own distinct region. The

error analyzed at each step is found through the Gini index, which is defined as:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}), \tag{3}$$

where $\hat{p}_{mk}$ is the fraction of the k^{th} class found with the m^{th} region. In certain situations, it may be beneficial to use an alternate method called cross-entropy, or log loss function. In its simplest form, it is defined as:

$$D = \sum_{k=1}^{K} \hat{p}_{mk} log(\hat{p}_{mk}). \tag{4}$$

Numerically, the Gini index and cross-entropy produce similar results, and which is used is dependent on the situation at hand. For this research, the Gini index was used for the decision tree models, and cross-entropy was used in the ANN discussed in the next section [13].

Computationally speaking, it is not feasible to look at every possible combination of decision tree splits and choose the one that has the lowest Gini index overall. Instead, the Gini index is calculated for each split individually. This top-down approach can suffer from high variance. This means that different subsamples of the data could produce very different results. In order to get around this and lower the variance at the expense of bias, a random sampling (with replacement) is taken from the data and a tree is grown from this sampling. This is repeated many times and the output of all the different trees is aggregated, which produces a much lower model variance and a much more accurate prediction tool. This is referred to as bagging, or random forest [13].

In order to lower the variance further at the expense of more bias, more randomness can be added into the model. Instead of searching for the split in each region that minimizes the Gini index, a random split can be made. This is called Extremely

Randomized Trees, or more commonly, Extra Forest. It is not easy to predetermine whether a random forest or extra forest will perform better. It will be different from one dataset to the next, and depends heavily on the bias and variance produced from each model. However, similar datasets will likely have consistency in which model performs better. Both of these models can be initialized and the results compared using Scitkit-Learn [12, 16].

Aggregating predictions from multiple models is a common practice with the intent to increase overall accuracy. Various linear models and tree based models can be pipelined into a final voting classifier. Using a simple majority-vote-wins classifier can pull from the strengths of all input models and produce a classifier that is more accurate than each model individually. However, the assumption here is that each model uses predictors that are as independent from one another as possible. If each model makes similar mistakes, then the voting classifier will simply propagate those mistakes and no improvements will be seen. Testing the effectiveness of a voting classifier on a specific dataset is a built-in feature of Scikit-Learn [12, 16].

2.4 Artificial Neural Networks

In recent years, ANNs have been extensively studied for use as advanced prediction tools. The effectiveness of these techniques has been shown in many different machine learning tasks, such as pattern recognition, natural language processing, event prediction, and many others. Also, they can be used with a high degree of success on many different types of data from images and video to stock market numbers and molecular compounds [7, 17, 18, 16, 19]. However, the term ANN is vague and covers a wide spectrum of machine learning models. Different types of data will require different approaches in order to make accurate predictions. One such approach is one-shot learning. The original paper on Siamese networks discusses one-shot learn-

ing and how a neural network can be trained to rank similarities between inputs [20]. It discusses training a network using a large dataset that is different from, yet similar to, another dataset with a limited number of samples. The term one-shot comes from the trained network looking at these new samples once and still being able to make accurate predictions. The Omniglot dataset was used in that research, which is a set of hand-drawn images of characters from 50 different alphabets. The success of that research was evidence that the entire range of possible classes is not needed during training. If an appropriate representation of classes is used, then if new, previously unknown classes are run through the Siamese network a clear distinction can still be made. The issue, of course, is that the elemental assay data for the samples is distinctly different than any image file. With the relative newness of Siamese networks, there has been limited research done on this topic. Though like other ANNs, it is assumed that a Siamese network can be adjusted to work efficiently with the data for this research.

The idea behind ANNs is that, like the human brain, there a many different nodes that are all acting upon the input data. Each node by itself will not make a good prediction, but when grouped together with several hundred, or even thousands of different nodes, the prediction capabilities can be substantial. More nodes generally mean better results, but this leads to much higher computational cost. The nodes are organized into layers. The data is fed into the first layer where each node acts upon each x value. Then the output of this first layer is fed into the second layer where again, each node acts upon each input. The output of the final layer represents the predicted y value. This concept is shown in Fig. 3. For Siamese networks, a single input is observations with the various features. One is a known observation, and the other is an unknown observation. For this research, the input is two particle observations with the various elemental assay features. The output layer of a binary

classification system can be a single node that gives a value between zero and one. In the case of a Siamese network, the closer the output is to one, the more similar the two observations are, and the closer the output is to zero, the less similar the two observations are [7, 20].

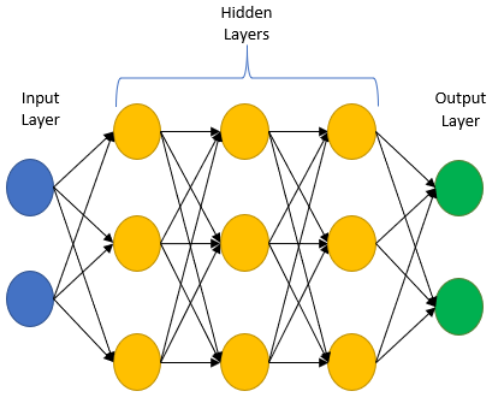

Figure 3. A simplified Artificial Neural Network with an input layer, two hidden layers, and an output layer.

Each node in an ANN has a bias associated with it, and there is a weight associated with each input into the node. Both the weights and the bias are applied to the input observation such that

$$Sum = \sum_{n=1}^{N} [\omega_n * x_n] + \beta, \tag{5}$$

where ω_n represents the weight associated with each feature, x_n, of the input observation, and β is the node bias. This sum is then fed into an activation function. There are a number of different activation functions known to be effective, and two common activation functions used for their simplicity and reliability are the ReLU and sigmoid functions [7]. ReLU stands for rectified linear unit and is the maximum value between zero and the input values, and is a piecewise linear function described by the function

$$ReLU = max(0, x). \tag{6}$$

This will produce a continuous, monotonically increasing line for all x in range $(0, \infty]$ and a continuous horizontal line at zero for all x in range $[-\infty, 0]$. The sigmoid activation function is

$$S(x) = \frac{1}{1 + e^x}, \tag{7}$$

which increases monotonically from zero to one as the input (x) goes from negative infinity to positive infinity. Because all negative inputs will produce an output of 0, the ReLU function tends to be less computationally intensive than the sigmoid function; however, the sigmoid function produces results more easily attributable to a probability. Both of these activation functions were used for this research.

A collection of nodes operating together at a specific depth within the neural network is called a layer. There are many types of different layers that can produce different effects. A dense layer is when all of the nodes in one layer are connected to all of the nodes in the next layer. A dropout layer is when a certain percentage of the nodes' outputs are randomly set to zero, which essentially makes the current layer be treated like a layer with a different number of nodes and connectivity to the prior layer. This has the effect of reducing overfitting in a model [7, 17]. Using Tensorflow, this can be set up so the dropout only occurs during training and all nodes during testing are still used. Other layers that Tensorflow provides include a flatten layer, which flattens multidimensional inputs into a one dimensional tensor, and a lambda layer, which provides a means for creating custom layers, similar to the lambda function in base Python [19].

Before the first training cycle, or training epoch, the weights and biases are randomly initialized. During consecutive epochs, these are updated based on how well the ANN performed. The specific metric that is back propagated through the network is the loss, and this is calculated through a specific loss function. There are many different loss functions that can be used for different situations. One that has shown

to be effective in conjunction with a Siamese network is the triplet loss function [21]. The biggest difference here is that instead of looking at the similarity between two observations, the triplet loss function compares three observations: One is the reference, or anchor; another is a positive match to the anchor; the final one is a negative match to the anchor. The triplet loss function takes vector encodings produced by the neural network and compares the vector distance between the anchor and positive observations, and the anchor and negative observations. The anchor-positive should have a small vector distance, whereas the anchor-negative should have a larger vector distance. An extra value is added to the loss to ensure that there is a significant margin, m, between the vector distances. This is done because the network could output everything as zero, then the loss would be small even though an accurate prediction has not been made. The loss is then calculated by:

$$d_1 = dist(A, P)$$
$$d_2 = dist(A, N)$$
$$loss = max(0, m + d_1 - d_2).$$

If d_2 is significantly large, then $m + d_1 - d_2$ could be negative. A negative loss does not add any value to the system, which is why the maximum between this sum and zero is taken [21].

With triplet loss, there is the possibility that the vector distance d_2 is smaller than the vector distance d_1. This implies that the negative is more similar to the anchor than the positive, which could make it more difficult for the network to train on. However, if the network manages to differentiate between the anchor and this 'hard' negative, the result could be a much more accurate network after training has finished. If a network is trained using only batches of hard examples, then this is

referred to as batch hard, whereas using all available combinations is referred to as batch all. There is also batch semi-hard, where d_2 falls somewhere between d_1 and $d_1 + m$. It has been shown that in certain circumstances, training a network using batch hard or semi-hard can lead to much greater accuracies overall [21].

A loss function used in contrast to triplet loss is binary cross-entropy, also known as the log loss function. Cross-entropy was introduced in Eq. 4 in the previous section. Binary cross-entropy is simply cross-entropy with only two possible outcomes, in this case either same or not-same [13]. Both binary cross-entropy and triplet loss were used for this research.

When making a prediction, a Siamese network will only have two options: same or not-same. The number of times a network makes a correct prediction will give the accuracy. However, this does not always give a clear indication of how well the model performs. Because of this, it is important to look at the results in the form of a confusion matrix in order to get a better idea at how the model is performing. The four basic components of a confusion matrix are true positive, true negative, false positive, and false negative results. True positive is the number of positive results predicted as positive and false negative is the number of positive results predicted as negative. Then true negative is the number of negative results predicted as negative, and false positive is the number of negative results predicted as positive.

What someone considers a positive result is purely subjective. For example, when looking at laboratory results on whether someone has a disease or not, it is often easier to think of the positive result as someone having the disease and the negative result as someone not having the disease. The positive does not need to represent something that is present, but should represent the target of interest. For example, for the purpose of trying to find people who are naturally resistant to a disease, it may be more beneficial to have the target be those who do not have the disease. The positive

and negative results simply need to be explicitly defined and remain consistent [13]. Once the actual and predicted values have been established, the results can be totaled and organized into a confusion matrix like the one shown in Table 1. This example shows that there are 30 negative observations and 10 positive observations.

Table 1. Example Confusion Matrix.

		Actual		Total
		Positive	Negative	
Predicted	Positive	8	7	15
	Negative	2	23	25
Total		10	30	40

From the confusion matrix, additional performance parameters can be calculated. These include precision, recall, and F_1 score. These three metrics are defined as follows:

$$\text{precision} = \frac{\text{true positive}}{\text{true positive } + \text{ false positive}}, \tag{8}$$

$$\text{recall} = \frac{\text{true positive}}{\text{true positive } + \text{ false negative}}, \tag{9}$$

$$F_1 = 2 \times \frac{\text{precision } \times \text{ recall}}{\text{precision } + \text{ recall}}. \tag{10}$$

In other words, precision is the number of accurately predicted positives over the total number of predicted positives, or the fraction of predicted items that are relevant. Recall is the number of accurately predicted positives over the total number of actual positives, or the fraction of relevant items that are predicted. The F_1 score is the harmonic mean of precision and recall and can give an idea of model performance with a single value. When multiple classes are present, these metrics can be calculated individually for each class. Then a macro-average for each metric can be obtained by

taking the arithmetic average from each class. A weighted average can be obtained in the same fashion. A final metric often used is called the micro-average, which reduces simply to the overall model accuracy [13].

When predicting if two observations from this data are the same, it is just as important and informative to know if the two observations are same or not-same, so positive and negative can be replaced with same and not-same for each sample. This categorization method is common when there is no clear target. However, the methods used to obtain the precision, recall, and F_1 score remain the same, but are now extended to account for multiple categories.

2.5 Moment Transformation and Load Balancing

Siamese networks were developed with the idea in mind that single observations are representative of the class they represent. When looking at the data for this work, it can not be guaranteed that a single observation is representative of the sample it belongs to, so a means for consolidating many observations into a more representative meta observation is needed. There are a number of different available options that can accomplish this, but as a proof of concept a moment transformation, using the first four moments of statistical distribution, was applied to the dataset.

In statistics, the population of a dataset can be summarized by the moments. The nth moment of a dataset about zero is the expected value of the nth power of a random variable, or

$$E[X^n] \tag{11}$$

In order to calculate the moments, a moment generating function can be used, and

is defined as (for a discrete distribution):

$$M_X(t) = E[e^{tX}] = \sum_x e^{tx} p(x), \tag{12}$$

where $p(x)$ is the probability of x. Then the nth derivative with respect to t is evaluated at $t = 0$ to get the nth moment. These equations are derived with the idea that the data follows a Gaussian distribution centered around zero. These equations become more complex when evaluating a dataset about a non-zero number, but the concept is the same [22].

The moment generating function can generate as many moments as needed to fully describe a distribution, but typically the first four moments are sufficient to give a unique summary of a distribution. The first moment is the mean and simply represents the average. The variance, skewness, and kurtosis are the second, third, and fourth centered moments, called this because their values are always in relation to, or centered about, the mean. The variance represents the spread of the data about the mean. Skewness represents how symmetrical a dataset is about the mean. Kurtosis represents how fat the tails of the distribution are, or how similar a dataset is to a normal distribution [22].

Performing a moment transformation on the data has the benefit of also balancing the dataset. A class that had 50 observations and a class that had 300 observations will both now have a single meta observation that will describe the class to some degree. A random sampling of each class can be taken multiple times in order to generate multiple meta observations. The extent to which these meta observations accurately describe the class is a part of this research. Alternatively, more traditional load balancing techniques can be used. Synthetic Minority Over-sampling Technique (SMOTE) uses a form of interpolation to generate artificial data that falls within the bounds of the real data [23]. It has also been used to load balance datasets similar

to the datasets used in this research [24].

2.6 Data Analysis

Nuclear data often refers to the isotopic information within a collected sample, but due to the relative ease that elemental information can be obtained, there has been a push for studying the practicality of using elemental information as nuclear predictors. As discussed in Chapter 1, each collected sample can provide a distinct fingerprint of the sample origin. Uranium, plutonium, and other actinides are obvious elemental features for determining sample origins, but there are also non-fuel elements that can be used in establishing a nuclear fingerprint. An example of this is zirconium which has highly desirable mechanical and nuclear properties for cladding nuclear fuel pellets to be used in a nuclear reactor. Hafnium on the other hand is a neutron poison and its removal from a nuclear system is preferred. Naturally, zirconium and hafnium are found together in similar quantities, so finding an extremely low relative abundance of hafnium compared to zirconium can be an indication of material processed for use in a nuclear reactor [25].

During the mining or milling processes of the nuclear fuel cycle there is still a large amount of non-fuel elements present within any given sample. The rare earth elements have been studied extensively as they can be used as effective geographical signatures that can link a sample to a specific uranium mine [4]. This has been done by looking at the elements individually and comparing them to elemental soil samples [5], and by taking a holistic approach using principle component analysis to compare mineral concentrations [26].

Traditionally, the data analysis has been done directly by human analysts, but this can be a time consuming process. For this reason, there has already been work done towards creating a machine learning program to accurately predict sample origins.

Work by Gum has shown that an algorithm can group observations into appropriate samples with a success rate greater than random chance [24]. Gum used various machine learning models including K Nearest Neighbors, Linear Discriminate Analysis, Decision Tree Classifier, Random Forest Classifier, and Bernoulli Naive Bayes, which were implemented through the Scikit-learn Python package. However, that research used the physical morphology of a particle for characterization rather than the elemental composition. A total of 43 different samples were used and the models had test accuracies between 5.8% and 19.2%, the lowest being the Decision Tree Classifier and the highest being the Bernoulli Naive Bayes. Random chance would be $\frac{1}{43}$, or about 2.3%, so every model tested proved to be better than random chance.

Work by Holland showed that machine learning is a viable means for predicting serial codes based on elemental assays [27]. That work explored different types of regression models, boosted decision tree models, discriminant analysis, and even some neural networks. Discriminant analysis did not initially perform well, and there was concern with the potential black-box nature of neural networks. Because of these reasons, Holland's work mainly focused the regression models and decision tree models. Ten different samples were used and the models achieved test accuracies between 64.6% and 86.5%, the lowest being a discriminate analysis technique and the highest being the C5.0 Decision Tree algorithm. That research showed that traditional machine learning techniques can be tuned to produce reliable prediction tools.

The issue with the previous work is that it focused on predicting the serial code associated with the sample, but the serial code is an arbitrary designator that does not say anything about sample origin. Using a Siamese network could give a more meaningful prediction capability.

III. Methodology

The first part of this research uses traditional machine learning techniques to predict which sample an observation comes from based on the relative elemental abundances of that observation. This will be used as the preliminary analysis to set a baseline for the the second part of this research, which will use an ANN to predict whether an unknown particle observation is similar to a particle observation from a known sample based on an invariant transformation of the all the observations within a sample. The ANN will be a Siamese Network using triplet loss to perform one-shot learning.

3.1 PreProcessing

The data used for this research came from a number of different projects. Each project is denoted by a unique project ID number. There are two distinct groups of projects: one where the data was collected using an SEM-EDS and another where the data was collected using an EPMA. The most notable differences between the two methods is the number of particle observations obtained, and the information contained within each particle observation. As discussed in Chapter 2, the TESCAN Mira-3 can make thousands of different particle observations, but each observation is not necessarily representative of the whole sample. In contrast, the E-Probe data has relatively few observations, each representative of particles of interest that were isolated by a technician. There are a total of 147 different samples between the two data sets. Table 2 summarizes the data obtained using each machine [28]. More detailed information on each dataset can be found in Appendix A.

Table 2. Data provided by the Air Force Technical Applications Center. Measurements conducted by the McCrone Group.
***This represents the total number of unique elemental features and each project will have overlapping features with other projects.**

Dataset	Project	Sample IDs	Particle Observations	Elemental Features
TESCAN	2017-162	3	20,221	36
	2018-204	7	58,243	39
Subtotal	2	10	78,464	40*
EPMA	06-111	3	43	20
	08-201	14	232	40
	10-065	2	36	26
	10-066	2	33	31
	10-068	1	15	8
	10-088	10	71	34
	11-031	12	214	29
	12-086	3	49	28
	12-089	1	28	20
	12-216	7	25	33
	12-221	7	79	34
	13-241	9	101	31
	15-066	2	31	30
	15-174	5	39	38
	15-210	6	47	37
	16-083	4	36	26
	16-224	14	155	48
	17-049	8	102	39
	18-034	4	49	40
	18-131	8	107	41
	18-158	4	40	32
	18-167	2	20	17
	18-227	1	10	8
	18-246	6	143	28
	19-312	2	43	39
Subtotal	25	137	1748	55*
Total	27	147	80,212	55*

The data from the TESCAN came in an Excel spreadsheet with each row being a single particle observation, and each observation having around 100 columns of data. Of these 100 columns, around half are the elemental assays, two establish a unique observation identification number (serial number and particle number), and the rest are additional details that further describe each observation that are not relevant for this analysis. The data from the JXA-8200 came in an Excel spreadsheet with each row being a single element from a single observation and 30 columns of information. Most of the columns correspond to sample processing, one column is for the serial number, one is for the element, and four are for the weight and atom percent and their percent errors. The weight percent was used in the analysis as it coincides with the data from the TESCAN. During the preliminary data processing the unnecessary information from each dataset was removed. Oxygen content was estimated by both machines, however the TESCAN data was not normalized after the addition of oxygen. Normalizing the TESCAN data was also done during pre-processing. It is never guaranteed that the same elements will be measured between different samples. In observations that do not contain a measured element that is present in other observations, the missing elemental feature is added with a corresponding value of zero. This ensures that each observation has the same number of features as every other observation.

The Python code used offered a way to shuffle the data the same way during every run in order to obtain consistent results between runs. This was accomplished using the model_selection.test_train_split function within the Scikit-Learn Python package [12]. The chosen value for the random seed parameter is arbitrary as long as that value remains constant. The variable that Scikit-Learn uses as this random seed is called random_state. The value used for the random_state variable, and all consecutive functions, was 7.

Training and validation sets were created from the all the data. For one run, the TESCAN data was split into 60% training and 40% validation, then the EPMA data was used as a second validation set. For the other run, the EPMA was split into 60% training and 40% validation with the TESCAN data used as the second validation set.

It is difficult to ascertain whether a single element will be relevant in predicting a sample once any potential learning algorithm is applied to a much more comprehensive database. Because of this, it was assumed that all elements could very well play a role in future predictions, so the only feature reduction technique used was Principle Component Analysis (PCA). Results from this could give a preliminary idea whether data reduction is possible. This was accomplished using the decomposition.PCA function within the Scikit-Learn Python package.

3.2 Preliminary Work

There are many well documented machine learning algorithms that are found within the Scikit-Learn package. The models chosen for the preliminary analysis were linear discriminate analysis, logistic regression, decision tree, random forest, extra forest, and voting classifier. The data was run through each model, varying certain hyper-parameters until the most desirable results were achieved. The parameters not adjusted were kept as the default values of the function. Once the optimal parameters were achieved, a final run was performed.

Three parameters in the linear discriminate analysis were specified. The shrinkage parameter was set to auto. This helps when the number of observations for a given sample is small compared to the number of features. The solver parameter was set to eigen as it supports the shrinkage parameter and can solve classification problems. The default value for the tol parameter is irrelevant as this is only used when the

solver is set to svd. There was no random_state parameter. Minimal optimization was performed to set these parameters.

The logistic regression function has six parameters that were specified. The parameter class_weight was set to balanced. This gives an inversely proportional weight to each sample based on the number of particle observations. Each sample is assumed to be equally relevant, so a class_weight parameter set to balanced helps to compensate the imbalanced representation of samples seen within the data. The multi_class parameter was set to multinomial and the solver parameter was set to lbfgs. These two together can provide faster convergence and better model calibration [12]. The C, tol, and random_state parameters were set to 300, 0.05, and 7, respectively. The C parameter is the inverse of the regularization parameter and was obtained by re-running the model until the highest training accuracy was obtained. Having a tol parameter too low can lead to a model overfitting on the data, so the tol parameter was increased between runs until the training accuracy began decreasing.

The decision tree, random forest, and extra forest all have similar parameters. The class_weight parameter was set to balanced, random_state was set to 7, and max_depth was set to 22. For random forest and extra forest, n_estimators was set to 50, max_features was set to None to ensure use of all features, and n_jobs was set to -1 to speed up run time.

To achieve the optimal value for max_depth, a single tree model was run 100 times with a depth starting at one and increasing the depth by one for every consecutive run. Then the accuracy for each was plotted versus the depth of the tree. The depth at which the accuracy began to asymptote was then chosen as the optimal max_depth value. A similar approach was used to obtain the optimal value for n_estimators. Using the value obtained for max_depth, a random forest model and extra forest model were each run 100 times starting with one tree and increasing the number

of trees by one for every consecutive run. Then the number of trees at which the accuracy began to level off was chosen as the optimal n_estimators value. All other parameters were kept constant during these runs.

The final model used was the voting classifier. The voting parameter was set to soft to better address the imbalance in the number of particle observations per sample, and n_jobs was set to -1. The voting model was run three times once all other parameters were set. Once with all models contributing, once with the decision tree, random forest, and extra forest, and a final time with only the forest models.

Finally, the results of all models were compared. The measure of success for every model was based primarily on the recall and weighted average scores, though all outputs related to precision, recall, and F_1-score were considered. For a given sample, precision represents the accuracy of the predictions, recall represents how often a given sample is predicted accurately, and the F1-Score is a combination of the two. The accuracy output parameter is the F1-Score calculated from looking at all the samples together, macro average is the arithmetic mean, and weighted average is based on the number of observations per sample. All of these parameters were obtained using the function metrics.classification_report in the Scikit-Learn Python package.

A final output parameter was the feature importance which ranked the elements by how much they contributed to the model accuracy. This, in conjunction with PCA, was used to determine the best features to use in the case of utilizing feature reduction techniques. This was done by plotting the explained variance ratio against the number of features, then taking a number of features that would still maintain a minimum explained variance ratio.

3.3 Siamese Network

In order for the data to be appropriately batched and run through the Siamese network, the data first needed to be load balanced so that every sample had the same number of observations. The Imbalanced-Learn Python package offers various techniques for oversampling, undersampling, and a combination of the two [29]. Undersampling would produce samples with too few observations, and the combination techniques proved to take an unreasonable amount of time considering this was not the focus of the research. Because of these factors, the oversampling SMOTE algorithm was used to increase the number of particle observations each sample had to a final load-balanced level of 19,013 observations per sample. The limitation here was that not all samples met the minimum requirement of six observations for SMOTE to work. For this reason, and to increase statistical variations of the final set of observations, all samples with less than ten observations were removed from the dataset. This reduced the number of EPMA observations from 137 to 87. A list of samples eliminated from further analysis can be found in Appendix B.

In order to transform the data, the observations from each sample were split into ten roughly even groups, 1,901 observations for nine groups and 1,904 observations in the remaining tenth group. Within each group, the first four moments were taken for each element. This resulted in a 4 by N matrix for each group, where N is the number of elements represented in the sample. These groups of elemental moments can be thought of as meta observations that more holistically represent the sample.

For half of the runs performed, the data was transformed using the first four moments of statistics. In order to give the network more variation, the observations within each sample were randomly split into ten even groups. The moment transformation was then performed on each individual group. This resulted in ten meta observations for each sample for all 97 (87 from EPMA and 10 from TESCAN)

remaining samples, totaling 970 moment transformation observations. For this transformed data, the same training and validation split was performed, where one run was performed with the TESCAN data used as the training and validation sets with a 60-40 split and the EPMA data used as a second validation set, then another run where the EPMA data was used for training and validation and the TESCAN used for a second validation.

In order for the network to be trained properly, the data needs to be batched in specific ways. For a Siamese network without triplet loss, there needs to be pairs of random observations that are either from the same sample or not from the same sample. Using triplet loss, there needs to be three observations; the anchor, the positive, and the negative. However, during testing of triplet loss, the batches need some additional randomization so the network can predict either same or not-same. During all batching processes, assurances were made so that a single observation is never paired with itself, that two different observations represent the same sample when there is supposed to be a same comparison, and that two different observations represent different samples when there is supposed to be a not-same comparison.

To accomplish this, the batching algorithm first creates a target array with zeros representing pairs of observations that are from the same sample, and ones representing pairs of observations that are from different samples. Then an anchor array of random integers between zero and n is created, where n is the number of available samples. A reference array is then created that contains the same integers as the anchor where the target array contains zeros, and different integers where the target

array contains ones. For example, if:

$$\text{Anchor, } A = \begin{bmatrix} 5 \\ 7 \\ 3 \\ 4 \end{bmatrix}, \textit{ and } \text{Target, } T = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \tag{13}$$

then a valid reference could be

$$\text{Reference, } R = \begin{bmatrix} 2 \\ 7 \\ 9 \\ 4 \end{bmatrix}. \tag{14}$$

This was done by first generating an array filled with random integers between one and n, then multiplying it element-wise by the target array. This array was then added to the anchor array. The final reference array was obtained by dividing by n and taking the remainder. For this example, if $n = 10$, then

$$\text{remainder} = [(T \times \text{randint} + A)/n] = \frac{\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} 7 \\ 1 \\ 6 \\ 9 \end{bmatrix} + \begin{bmatrix} 5 \\ 7 \\ 3 \\ 4 \end{bmatrix}}{10} = \frac{\begin{bmatrix} 12 \\ 7 \\ 9 \\ 4 \end{bmatrix}}{10} = \frac{\begin{bmatrix} 2 \\ 7 \\ 9 \\ 4 \end{bmatrix}}{} = R \tag{15}$$

This method ensures that when two observations are supposed to be from the same sample, the result is always $0 \times \text{randint} + A = A$. When two observations are supposed to be from different samples, the result will always be $1 \times [\text{int} > 0] + A \neq A$. These

anchor and reference arrays were used as indexing arrays to obtain the appropriate samples from the data. An index array of random integers between zero and k was then created, where k is the number of observations within each sample. This index array was used to choose a random observation from within a sample, and was created in the same fashion as the anchor and reference arrays.

This method was performed during training only. When testing, every same/not-same combination was sent through the model, only randomizing the specific observation chosen within a sample. This ensured that each sample was paired with every other sample, which forced the model to make predictions on how similar each sample was to one another.

As this research is meant to be a proof of concept, optimizing the network itself was not a priority, though various network depths and widths were explored to obtain a model that performed consistently well. The moment transformed data resulted in a two-dimensional 4 by N matrix, where N is the number of elements, so the first layer was a flatten layer to reduce the dimensionality. This ensures that the moment transformed data could successfully pass through the consecutive layers, and should have no effect on the non-transformed data. The network architectures used are summarized in Tables 3 and 4.

Anchor				Reference			
Layer	Activation Shape	Activation Size	Trainable Parameters	Layer	Activation Shape	Activation Size	Trainable Parameters*
Input	(4, 58)	232	0	Input	(4, 58)	232	0
Flatten	(232)	232	0	Flatten	(232)	232	0
Fully Connected	(10)	10	2330	Fully Connected	(10)	10	0
Fully Connected	(15)	15	165	Fully Connected	(15)	15	0
Dropout	(15)	15	0	Dropout	(15)	15	0
Fully Connected	(20)	20	320	Fully Connected	(20)	20	0
subtotal			2815	subtotal			0

Layer	Activation Shape	Activation Size	Trainable Parameters
Lambda**		20	0
Fully Connected	(1)	1	21
Total			2836

Table 3. The basic architecture of the Siamese network used for this research.
***The number of trainable parameters for the reference branch is zero to ensure the weights are the same between the anchor and reference.**
****The Lambda function compares the anchor encoding to the reference encoding using an element-wise squared difference**

Positive			Anchor				Negative		
Layer	Shape	Trainable Params*	Layer	Activation Shape	Activation Size	Trainable Parameters	Layer	Shape	Trainable Params*
Input	(4, 58)	0	Input	(4, 58)	232	0	Input	(4, 58)	0
Flatten	(232)	0	Flatten	(232)	232	0	Flatten	(232)	0
Fully Connected	(10)	0	Fully Connected	(10)	10	2330	Fully Connected	(10)	0
Fully Connected	(15)	0	Fully Connected	(15)	15	165	Fully Connected	(15)	0
Dropout	(15)	0	Dropout	(15)	15	0	Dropout	(15)	0
Fully Connected	(20)	0	Fully Connected	(20)	20	320	Fully Connected	(20)	0
Total		0	Total			2815	Total		0

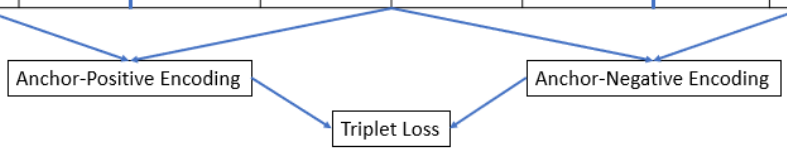

Table 4. The basic architecture of the triplet loss network used for this research.
***The number of trainable parameters for the positive and negative branches is zero to ensure the weights are the same between the anchor, positive, and negative.**

The Lambda function in the Siamese network takes the encodings from both the anchor and reference and performs an element-wise squared difference. Since this is a simple mathematical operation, there are no trainable parameters in this layer.

For the models with triplet loss, three separate batching processes were tested; batch all, batch hard, and batch semi-hard. Batch all factors in the loss from every group of triplets. Batch hard dropped all triplets where the anchor-positive vector length was greater than the anchor-negative vector length. Batch semi-hard kept only the triplets where the anchor-negative vector length fell between the anchor-positive vector length and the anchor-positive vector length plus the margin. The margin used for all runs was set to 1.

A model using triplet loss does not make definite predictions. It only outputs the encodings, and the triplet loss function only outputs a loss. In order for the model to make predictions when checking accuracy, a threshold parameter needed to be implemented. If the Euclidean distance between the anchor-positive encoding and anchor-negative encoding is above the threshold, then the two observation pairs are counted as different. If the Euclidean distance is below the threshold, then the two observation pairs are counted as the same. However, the Euclidean distance can have a wide range of values, which makes choosing a single threshold value that can accommodate all the input data somewhat difficult. To circumvent this issue, these values were normalized to cover a range between zero and one. This required the assumption that any distance grater than the margin value was caused by a two observations that are from different samples. If the calculated distance was above the margin value, then the distance was set equal to the margin. Then all values were divided by the margin. This gave values between zero and one, similar in fashion to the probability predictions made by the base Siamese network. Then the absolute difference between these probabilities and target predictions was taken. The smaller

the difference, the closer the prediction was to being accurate. So if this difference fell below the threshold, then the prediction was counted as correct. The threshold used for all runs was set to 0.5. This value is somewhat arbitrary and was chosen simply because it is the midpoint between zero and one. It can be considered another hyper parameter that can be adjusted by the end user, which will be discussed more in the next section.

A number of different runs were performed. Only a single factor was changed between runs in order to properly compare and contrast the effects of each change. The factors examined were the dataset used for training and validation (TESCAN versus EPMA), the application of moment transformation to the data, and triplet loss versus binary cross entropy. For each run performed, the loss and the accuracy of the training and validation sets were plotted versus the number of training epochs. Also, a confusion matrix was created based on predictions made after the model was finished training. Results from this confusion matrix were taken to calculate and plot precision, recall, f1-scores, and accuracy for each sample tested. A major goal for this research is to find a way to identify samples that are similar, so the overall accuracy of each run was analyzed, however a great deal of attention was paid to the incorrect predictions. When the EPMA data was provided, groups of samples were also given. The samples within each group were determined to be similar to some degree by the nuclear analysts who originally examined the data. The list of groups are shown in Table 7 at the end of this chapter, and will be used to determine how effective the model is at determining similarities between samples.

One main objective of this research is to move away from predicting the sample serial number based on a particle observation, and move towards a method of determining how similar samples are to one another based on particle observations. However, this creates an issue in how to label the data. If the assumption holds true

that all samples within a group are similar to one another, then ideally all of the particle observations within that group will be predicted as similar to one another. If the labels are kept as they currently are (each particle observation labeled with the sample ID it belongs to), then the accuracy will be lower than what is expected. For example, group 1 has three samples associated with it. If all the observations from that group are predicted as similar to one another, but each observation is labeled with the sample ID, then the accuracy that the model calculates will only be 33% at best. The model assumes that SN014855 is different from both SN014856 and SN014857, but will always predict all three as the same.

The alternative is to re-label every particle observation with the group it belongs to. However, this comes with its own set of issues. One major issue is that many samples do not fit neatly into a single group. They share attributes with many other samples. The mixed group from the EPMA is a group of samples that fit this description and account for approximately 35% of all of the samples used from the EPMA dataset. Having an 'other' category of samples would make it difficult for any model to zero in on what makes an 'other' sample belong to the 'other' category. Further, many of the 'other' samples share characteristics with samples from many of the other groups. This is part of what makes the mixed group mixed. This could lead to the model finding correlations in the data that have no physical meaning.

It was ultimately decided to keep each particle observation labeled with the sample it belongs to, not the group, with the expectation that accuracies will be lower due to the assumed similarities between samples. Keeping this in mind, the expected precision and recall values for both same and not-same predictions can be calculated. Looking at same predictions, precision is the accurate same predictions divided by the total number of same predictions. Looking at group 1 from the EPMA data, the precision for making same predictions is then 1/3. Again, because of how the particle

observations are labeled and the assumed similarities between samples, a model will make a same prediction for every possible pair of observations that are from group 1, but only one third of those same predictions will be accurate, according to the labels. This thought process can be extended to precision and recall for same and not-same predictions for all of the groups and projects in both the TESCAN and EPMA datasets. These expected values are shown in Tables 5 and 6.

Table 5. Estimated precision and recall for predicting both same and not-same, separated by sample groups from the EProbe Data.

Group	Number of Samples	Same		Not-Same	
		Precision	Recall	Precision	Recall
G1	3	0.33	1.0	1.0	0.98
G2	25	0.04	1.0	1.0	0.72
G3	12	0.083	1.0	1.0	0.87
G4	11	0.091	1.0	1.0	0.88
G5	2	0.5	1.0	1.0	0.99
Mixed	32	0.031	1.0	1.0	0.64
Macro Avg		0.18	1.0	1.0	0.85

Table 6. Estimated precision and recall for predicting both same and not-same, separated by sample projects from the TESCAN Data.

Group	Number of Samples	Same		Not-Same	
		Precision	Recall	Precision	Recall
G1	3	0.33	1.0	1.0	0.78
G2	7	0.14	1.0	1.0	0.33
Macro Avg		0.24	1.0	1.0	0.56

Table 7. Projects and samples grouped by similarity.

Group	Projects	Sample IDs
g1	06-111	SN014855, SN014856, SN014857
g2	08-201	SN016837, SN016838, SN016839, SN016840, SN016841, SN016842 SN016843, SN016844, SN016845, SN016846, SN016847, SN016848 SN016849, SN016850
	12-089	SN019688
	18-167	SN024980, SN024981
	10-068	SN017959
	10-065	SN017952, SN017953
	12-086	SN019683, SN019684, SN019685
	18-167	SN024980, SN024981
	18-227	SN024984
	18-246	SN025079, SN025080, SN025081, SN025082, SN025083, SN025084
g3	15-210	SN022456, SN022457, SN022458, SN022459, SN022460, SN022464
	16-083	SN022817, SN022818, SN022819, SN022823
	16-224	SN023170, SN023172, SN023174, SN023175, SN023176, SN023177 SN023178, SN023179, SN023180, SN023181, SN023182, SN023183 SN023185, SN023186
g4	11-031	SN018800, SN018802, SN018803, SN018804, SN018805, SN018806 SN018807, SN018808, SN018809, SN018810, SN018811, SN018812
g5	15-066	SN022017, SN022018
mixed	18-158	SN024822, SN024823, SN024824, SN024825
	19-312	SN026227, SN026228
	12-216	SN020087, SN020088, SN020089, SN020090, SN020091, SN020092 SN020093
	12-221	SN020100, SN020101, SN020102, SN020104, SN020105, SN020107 SN020109
	13-241	SN020885, SN020886, SN020887, SN020888, SN020889, SN020890 SN020891, SN020892, SN020893
	15-174	SN022312, SN022313, SN022314, SN022315, SN022316
	17-049	SN023522, SN023523, SN023524, SN023527, SN023528, SN023531 SN023535, SN023538
	18-131	SN024794, SN024795, SN024796, SN024797, SN024798, SN024799 SN024800, SN024801
	18-034	SN024399, SN024400, SN024401, SN024402
	10-088	SN018062, SN018064, SN018065, SN018066, SN018067, SN018068 SN018069, SN018070, SN018071, SN018072

IV. Results & Discussion

The main purpose of this work was to determine a way to identify samples that are similar to one another. A baseline was established by expanding upon previous work done with traditional machine learning techniques. Then using a combination of moment transformation and a Siamese network with triplet loss, a means of determining sample similarities was established. Prior to this work, the samples were sorted into groups based on the evaluations done by professionals within the field of nuclear forensics. The metric of how accurately the model grouped the samples will be discussed.

4.1 Traditional Machine Learning Techniques

The work with traditional machine learning techniques attempted to show that samples with few observations performed significantly worse than those with many observations. The voting classifier performed the best with an accuracy of 0.782. Within this model, the sample with the fewest observations had an f1-score of 0.148, whereas the sample with the most observations had an f1-score of 0.903. This trend was seen in every model currently used in this work.

4.1.1 Principle Component Analysis

Using Principle Component Analysis, the explained variance ratio was plotted versus the number of features shown in Fig. 4. It shows here that as few as 6 features can be used while still maintaining most of the information, specifically, 92.9% of the data set's variance still lies along the principle components [16]. The elements that were most frequently chosen as the most important are copper, uranium, oxygen, iron, silicon, and aluminum, though their order was not always the same. This is expected

to be highly correlated to the samples examined and will likely be different for more diverse samples. Also, the F1-Score of all the samples with less than 1000 observations was drastically reduced, with the logistic regression and linear discriminate analysis models not representing those samples at all.

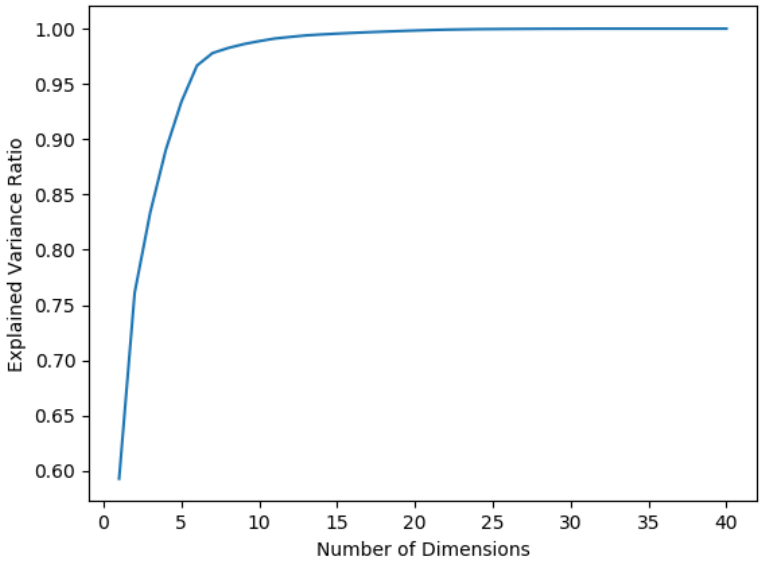

Figure 4. Expected Variance vs Number of Dimensions Using PCA on sample Data Set.

4.1.2 Linear Results

The output for the linear models with the best achieved accuracy is summarized in Table 8. These results reflect the best achieved hyper-parameters of $C = 300$ and tol $= 0.05$. Logistic regression appears to perform better with either the TESCAN data or EProbe data, not both. Linear discriminate analysis had a higher accuracy than logistic regression with the combined dataset. It is also apparent that for each scenario, the macro average was greatly below the accuracy and weighted average. This shows that each of the linear models has a difficult time accounting for an imbalanced dataset. However, separating the datasets does not solve the issues since the

EProbe dataset performed poorly overall yet accounts for about 90% of the samples but only about 2% of the particle observations.

Table 8. Precision, Recall, and F1-Score results for Logistic Regression (left) and Linear Discriminate Analysis (right).

	Logistic Regression				Linear Discriminate Analysis			
		Precision	Recall	F1-Score		Precision	Recall	F1-Score
EProbe	Accuracy			0.237	Accuracy			0.149
	Macro Avg	0.191	0.181	0.171	Macro Avg	0.121	0.120	0.098
	Weight Avg	0.283	0.237	0.241	Weight Avg	0.172	0.149	0.133
		Precision	Recall	F1-Score		Precision	Recall	F1-Score
TESCAN	Accuracy			0.615	Accuracy			0.610
	Macro Avg	0.430	0.524	0.409	Macro Avg	0.389	0.448	0.372
	Weight Avg	0.679	0.615	0.616	Weight Avg	0.605	0.610	0.578
		Precision	Recall	F1-Score		Precision	Recall	F1-Score
Both	Accuracy			0.498	Accuracy			0.587
	Macro Avg	0.082	0.181	0.087	Macro Avg	0.043	0.071	0.044
	Weight Avg	0.653	0.498	0.526	Weight Avg	0.597	0.587	0.566

4.1.3 Decision Trees

Next to be analyzed was a single decision tree. Figures 5, 6, and 7 plot the accuracy versus the depth of tree for both datasets, combined and separate. The red lines indicate where the maximum accuracy was achieved. For all three, there is an asymptotic trend reached above a depth of 40, though the TESCAN-only data levels off much sooner. This is likely due to large number of training observations found within the TESCAN data. The TESCAN-only data also reaches a higher accuracy rate. This is presumed to be a result of the model having difficulty finding a pattern with the low number of observations found within the EProbe data. Since the total accuracy does not change significantly above a depth of 40, this number was used for consecutive runs. The jaggedness of the plots is the result of the model trying to converge on a single accuracy value. This is exacerbated in early runs with the

combined dataset, likely because of the stark difference between the two datasets.

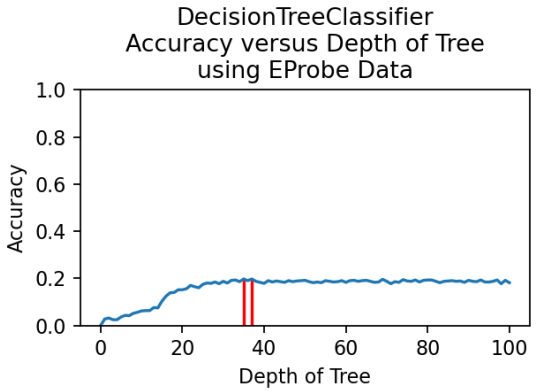

Figure 5. Accuracy versus depth of tree trained on EProbe data. The red lines show where the maximum accuracy was achieved.

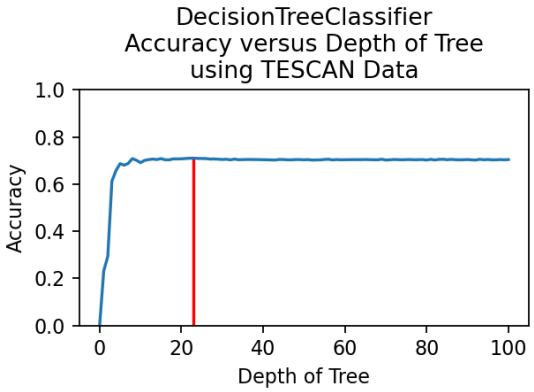

Figure 6. Accuracy versus depth of tree trained on TESCAN data. The red line shows where the maximum accuracy was achieved.

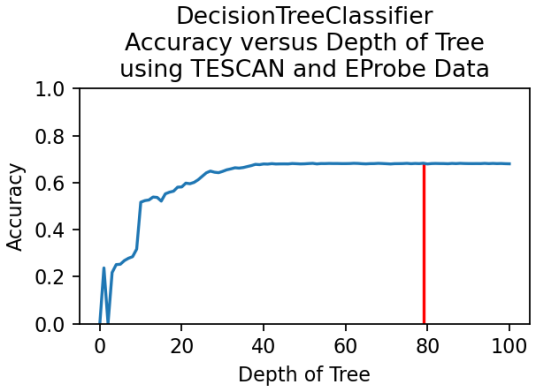

Figure 7. Accuracy versus depth of tree trained on both TESCAN and EProbe data. The red line shows where the maximum accuracy was achieved.

The results for the decision tree classifier are summarized in Table 9. The decision tree classifier showed similar trends as the linear models; poor performance with the EProbe data, moderate to high performance with the TESCAN data, and moderate to high accuracy and weighted average but low macro average with the combined dataset. Again, there is an apparent difficulty with the model dealing with imbalanced datasets. The logistic regression model had higher scores in every metric when trained

on the EProbe data compared to the decision tree. However, the decision tree classifier had notably better precision, recall, and f1-scores when trained on the TESCAN data or combined dataset. The decision tree classifier had completely better metrics compared to linear discriminate analysis for all dataset combinations.

Table 9. Precision, Recall, and F1-Score results for a Decision Tree Classifier.

	Decision Tree Classifier			
		Precision	Recall	F1-Score
EProbe	Accuracy			0.200
	Macro Avg	0.152	0.155	0.139
	Weight Avg	0.226	0.200	0.196
		Precision	Recall	F1-Score
TESCAN	Accuracy			0.704
	Macro Avg	0.502	0.498	0.500
	Weight Avg	0.705	0.704	0.705
		Precision	Recall	F1-Score
Both	Accuracy			0.680
	Macro Avg	0.167	0.176	0.157
	Weight Avg	0.686	0.680	0.682

The accuracy vs number of trees, for both the random forest and extra forest, classifiers is shown in Figures 8 through 13. Again it is shown that the TESCAN-only data performs better, though the combined dataset does has a comparable accuracy. Though the model chose peak accuracy at a number of trees less than 100, it appears there is an asymptotic trend as more trees are added. Because of this, and for the sake of computational cost, the number of trees selected for consecutive runs was 50. This resulted in a negligible loss of accuracy.

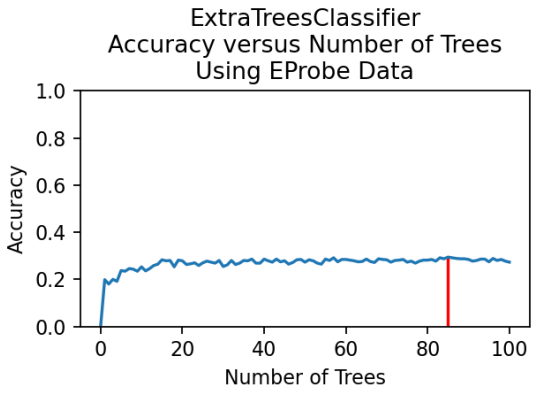

Figure 8. Accuracy versus number of trees for an Extra Forest classifier trained on EProbe data.

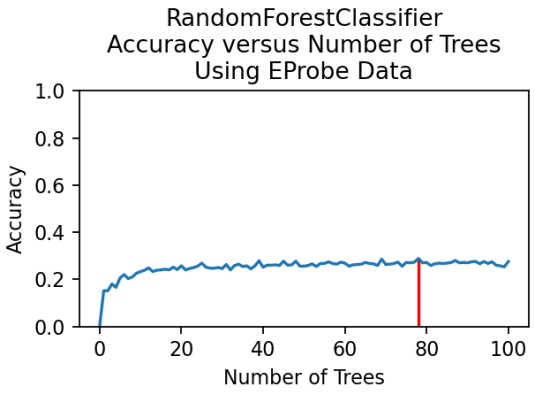

Figure 9. Accuracy versus number of trees for a Random Forest classifier trained on EProbe data.

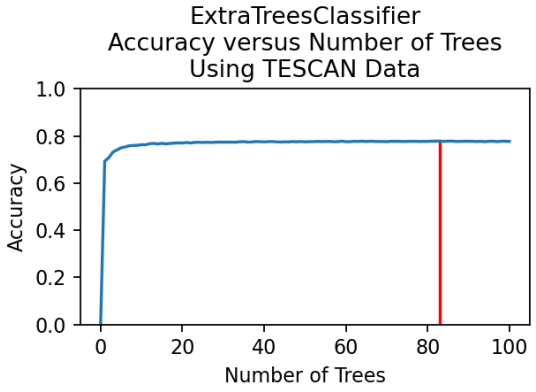

Figure 10. Accuracy versus number of trees for an Extra Forest classifier trained on TESCAN data.

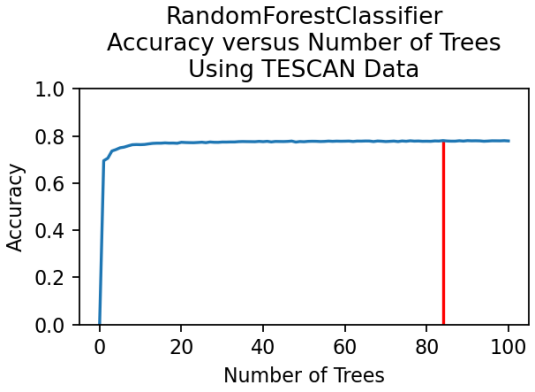

Figure 11. Accuracy versus number of trees for a Random Forest classifier trained on TESCAN data.

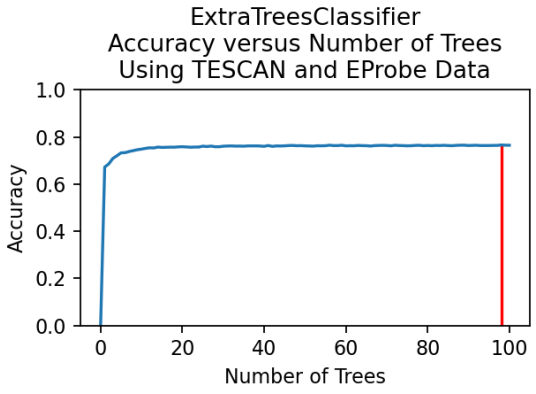

Figure 12. Accuracy versus number of trees for an Extra Forest classifier trained on both EProbe and TESCAN data.

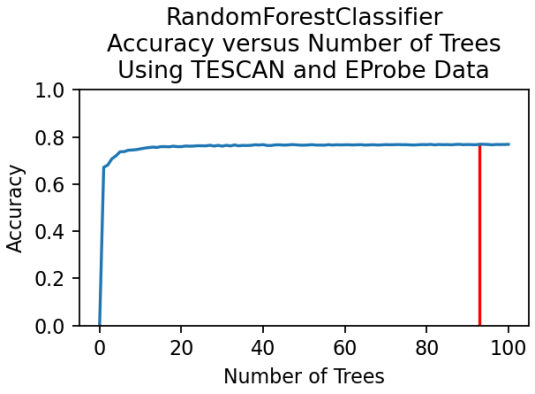

Figure 13. Accuracy versus number of trees for a Random Forest classifier trained on both EProbe and TESCAN data.

A side-by-side comparison of the final results for random forest and extra forest is shown in Table 10. This shows that both models have comparable results, though the extra forest classifier does appear to perform slightly better on the macro average. This indicates the extra forest model better accounts for the imbalanced nature of the datasets. This is likely due to the randomized decision boundaries of the extra forest classifier, which tends to lower the variance and increase the bias, giving the model a tighter fit around the data.

Table 10. Precision, Recall, and F1-Score results for Random Forest (left) and Extra Forest (right) classifiers.

	Random Forest				Extra Forest			
		Precision	Recall	F1-Score		Precision	Recall	F1-Score
EProbe	Accuracy			0.256	Accuracy			0.284
	Macro Avg	0.210	0.194	0.183	Macro Avg	0.205	0.206	0.192
	Weight Avg	0.299	0.256	0.253	Weight Avg	0.314	0.284	0.281
		Precision	Recall	F1-Score		Precision	Recall	F1-Score
TESCAN	Accuracy			0.778	Accuracy			0.775
	Macro Avg	0.586	0.540	0.553	Macro Avg	0.602	0.553	0.570
	Weight Avg	0.766	0.778	0.768	Weight Avg	0.765	0.775	0.766
		Precision	Recall	F1-Score		Precision	Recall	F1-Score
Both	Accuracy			0.765	Accuracy			0.763
	Macro Avg	0.253	0.217	0.209	Macro Avg	0.287	0.226	0.229
	Weight Avg	0.757	0.765	0.757	Weight Avg	0.757	0.763	0.756

4.1.4 Voting Classifier

The final model used was the voting classifier. Once the hyper-parameters of each individual model were tuned, the voting classifier was run with two different scenarios: all models contributing, and forest based models only. The results are shown in Table 11. The forest-only voting model performed very similarly to the extra forest model, and the small differences appear to be simple statistical deviations. The forest-only voting classifier performed marginally better than the voting classifier with all models.

Again, using only TESCAN data produced much higher macro average scores, and using the EProbe data reduced the macro average scores significantly.

Table 11. Precision, Recall, and F1-Score results for a Voting Classifier using all models (left) and forest models only (right).

	All Models				Forest Models Only			
		Precision	Recall	F1-Score		Precision	Recall	F1-Score
EProbe	Accuracy			0.243	Accuracy			0.283
	Macro Avg	0.182	0.194	0.171	Macro Avg	0.211	0.219	0.198
	Weight Avg	0.258	0.243	0.231	Weight Avg	0.308	0.283	0.275
		Precision	Recall	F1-Score		Precision	Recall	F1-Score
TESCAN	Accuracy			0.761	Accuracy			0.780
	Macro Avg	0.560	0.557	0.551	Macro Avg	0.611	0.557	0.574
	Weight Avg	0.752	0.761	0.752	Weight Avg	0.770	0.780	0.771
		Precision	Recall	F1-Score		Precision	Recall	F1-Score
Both	Accuracy			0.743	Accuracy			0.768
	Macro Avg	0.224	0.198	0.192	Macro Avg	0.261	0.223	0.221
	Weight Avg	0.738	0.743	0.734	Weight Avg	0.759	0.768	0.759

The precision, recall, and f1-scores for each sample for all of the models can be found in Appendix B. Each model tested had low macro average scores. This shows that a dataset with few observations per sample can lead to severe limitations in a model's prediction capabilities. There is a large amount of data similar to the EProbe data presented here, so it is currently impractical to rely on predictions using traditional machine learning techniques.

4.2 Siamese Network and Triplet Loss

Due to the limitations of traditional machine learning techniques, ANNs were examined. A Siamese network using triplet loss could be used to make more accurate predictions on data with few observations. There is also the concept of using a moment transformation to obtain a meta observation that represents the whole sample, which

could help a network learn what the data is supposed to look like. Four similar Siamese networks were constructed to test triplet loss and moment transformation. Binary cross-entropy was used as a baseline against triplet loss, and both the binary cross-entropy and triplet loss models were run with and without a moment transformation. Each combination was first trained on the TESCAN data and validated using the EProbe data. Then each combination was trained on the EProbe data and validated using the TESCAN data. This created a total of eight different trained models that were analyzed. The precision and recall that would be achieved through random chance alone is shown in Table 12. Keep in mind that during validation, each possible combination of sample pairs is sent through the model. This means there are far more pairs of observations from different samples, so the precision values will be skewed by how many different samples are in the dataset. For comparison, a summary of the precision and recall of all runs is shown in Tables 13 and 14. An important thing to remember when looking at these tables is that the sum of the precision for same and not-same predictions does not equal one. The precision for same predictions is the accurate same predictions over all of the same predictions. The precision for not-same predictions is the accurate not-same predictions over all not-same predictions. The numerator does not take into account any of the false positives or false negatives.

Table 12. The precision and recall through randomly predicting same or not-same given every possible combination of samples.

	Same		Not-Same	
	Precision	Recall	Precision	Recall
Validated on EProbe	0.012	0.5	0.988	0.5
Validated on TESCAN	0.1	0.5	0.9	0.5

Table 13. Precision and Recall for predicting both same and recall not-same validated using EProbe data.

	Same		Not-Same	
	Precision	Recall	Precision	Recall
Expected	0.18	1.0	1.0	0.84
Random	0.012	0.5	0.988	0.5
Binary Cross-entropy No Transformation	0.012	0.910	0.991	0.117
Binary Cross-entropy Moment Transformation	0.049	0.993	0.999	0.775
Triplet Loss No Transformation	0.016	0.860	0.996	0.378
Triplet Loss Moment Transformation	0.032	0.998	0.999	0.648

Table 14. Precision and Recall for predicting both same and not-same validated using TESCAN data.

	Same		Not-Same	
	Precision	Recall	Precision	Recall
Expected	0.24	1.0	1.0	0.56
Random	0.1	0.5	0.9	0.5
Binary Cross-entropy No Transformation	0.150	0.466	0.922	0.706
Binary Cross-entropy Moment Transformation	0.299	0.184	0.913	0.952
Triplet Loss No Transformation	0.143	0.663	0.937	0.558
Triplet Loss Moment Transformation	0.164	0.764	0.955	0.578

The big picture will be looked at first, this involves analyzing the overall accuracy of each model. This includes the training and validation accuracies and how they changed over the course of training. Then the Precision, Recall, Accuracy, and F1-Score (PRAF) of each sample within each model will be analyzed. Due to the large

number of samples, this will be visually represented as a bar graph, and tables with the actual values can be found in Appendix C. The final metric to be analyzed will be how similar each sample is to one another, as predicted by the each model.

4.2.1 Trained on TESCAN and Validated with EProbe

Figures 14 through 17 show the accuracy plotted versus the training epoch. These graphs imply that using triplet loss gives a much more stable model. The validation accuracy using binary cross-entropy varies and does not appear to be converging towards any single value. This was consistent across multiple runs. Using triplet loss consistently performed better than binary cross-entropy when no transformation was used. When the moment transformation was applied, triplet loss sometimes performed better and sometimes performed worse. Figure 17 is an example of the lower accuracy and Figure 18 is an example of the higher accuracy for a different run. Due to the computational time required for each run, not enough runs were performed to determine how often a model using moment transformation performed better than no transformation for triplet loss. However, if this could be stabilized, then a model using triplet loss with moment transformation would be the best performer in both accuracy and stability. One potential way to stabilize a network using triplet loss is to implement batch hard or semi-hard. It is recommended that any future research work towards implementing batch hard or semi-hard. This would help the network learn how to better differentiate the anchor-negative pairs that are very similar to the anchor-positive pairs.

Figures 18 and 19 show the accuracy and loss, respectively, of a different triplet loss and moment transformation run. It appears that the model is hitting a local minimum when optimizing the loss. This trend was seen during each run using triplet loss and moment transformation, though not always to the same degree. The margin

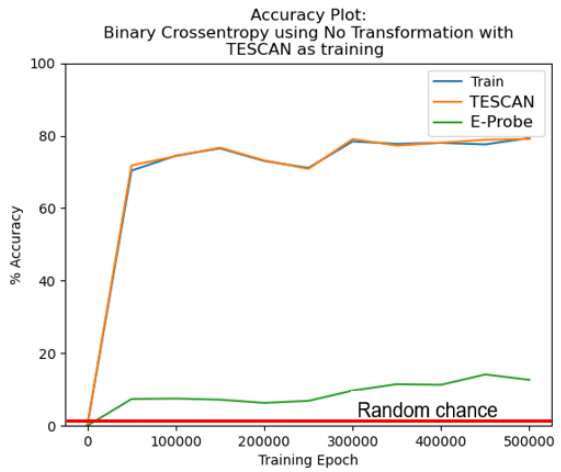

Figure 14. Binary Cross-entropy with no Transformation. Validated on TESCAN and EProbe Data.

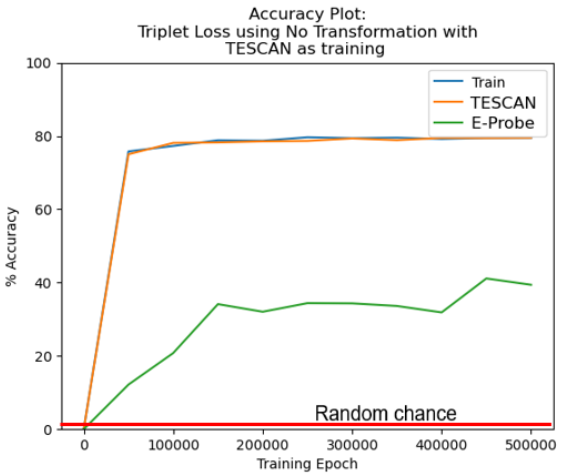

Figure 15. Triplet Loss with no transformation. Validated on TESCAN and EProbe Data.

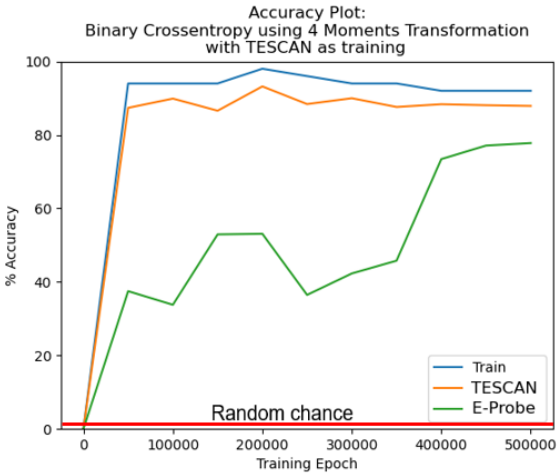

Figure 16. Binary Cross-entropy with 4 moment transformation. Validated on TESCAN and EProbe Data.

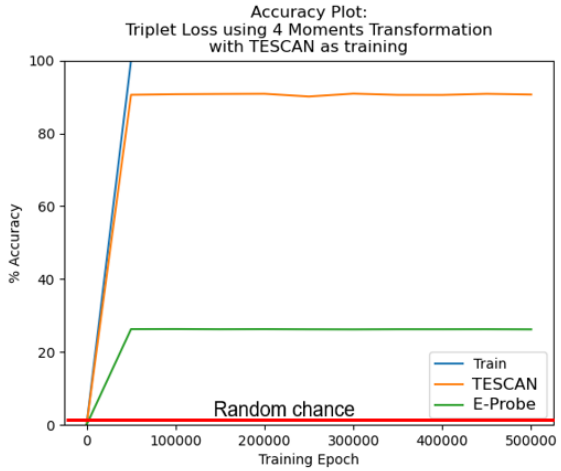

Figure 17. Triplet Loss with 4 moment transformation. Validated on TESCAN and EProbe Data.

for these runs was set equal to 1, so increasing this margin right from the beginning or gradually over time would likely increase the final performance of the model.

Before looking at the PRAF scores, it is important to have an idea of what the model should achieve. An ideal scenario would be one that achieves a recall close to 1 when predicting same and a precision close to 1 when predicting not-same. This would indicate two things: First, all sample pairs that are the same are always

predicted as the same; second, only sample pairs that are different would be predicted as not-same. The precision when predicting same and recall when predicting not-same would depend on how many different samples are similar to one another. The more underlying patterns amongst samples, the lower the precision will be when predicting same, and the lower the recall will be when predicting not-same.

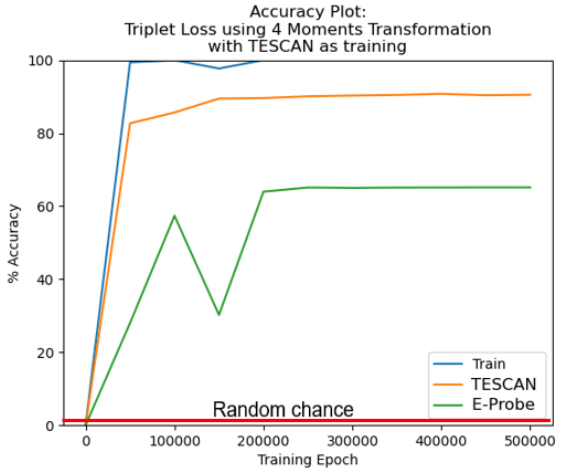

Figure 18. Accuracy of a model using Triplet Loss with 4 moment transformation versus training epoch.

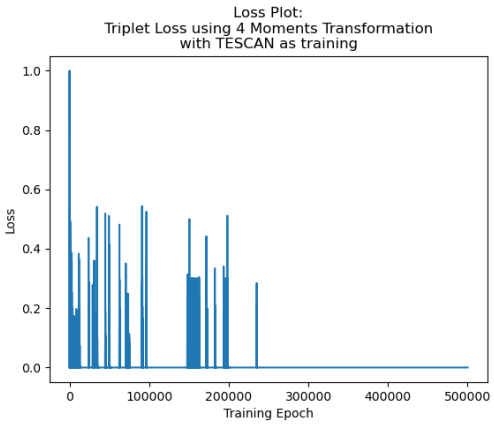

Figure 19. Loss of a model using Triplet Loss with 4 moment transformation versus training epoch.

It is important to remember that when the model is determining accuracy, it is comparing each sample to every other sample. This means that an accuracy similar to random chance would be around $\frac{1}{\text{Number of samples}} = \frac{1}{87} \approx 1.15\%$ (87 samples in the EProbe validation data). On the other hand, an accuracy that is too high implies that the model can distinctly identify each sample, which means it would be unsuccessful in grouping similar samples. 'How high is too high?' is not an easy question to answer as it depends greatly on what sorts of meaningful connections can be found within the data. With the six sample groups provided with the EProbe data, an initial target accuracy could be $\frac{5}{6} \approx 83\%$. Recall from the previous chapter Table 5, which shows the expected precision and recall for same and not-same predictions. It is broken up by the provided sample groups, and also shows the estimated macro average for the

whole model. Remember that precision for predicting same is defined as the number of correct same predictions over the total number of same predictions, and recall for predicting not-same is the the number of correct not-same predictions over the total number of samples that are not-same. There are two important take-aways here: as the number of samples in a single group increases, these precision and recall values will decrease; and is that as the number of sample groups increases, the expected model accuracy will decrease.

The PRAF scores for triplet loss using the moment transformation are shown in Figures 20 and 21. A thing to note when looking at these graphs is that these scores are absolute. They do not sum to 1. It may appear as if the scores are summing to 1, but this is not the case. For example, in Figure 20 the recall is approximately equal to 1.0 for every sample, and the accuracy for most samples is around 0.2, so the sum would then be around 1.2. This model combination is being highlighted for two reasons. First, as stated the recall for predicting 'same' is approximately 1.0 for every sample. This means the model always predicts correctly when it is presented with two particle observations from the same sample (i.e. always predicts same when the two observations are from the same sample). Put another way, the model was not confused with what was supposed to be the same. It still had difficulties pointing out which samples were similar to other samples, but it was always certain when two observations represented the same sample. The second reason this model is being highlighted is that the precision for predicting 'not-same' is approximately 1.0 for every sample. This means the model may not always predict not-same, but when it does, the two samples are different, i.e. the model always makes accurate not-same predictions.

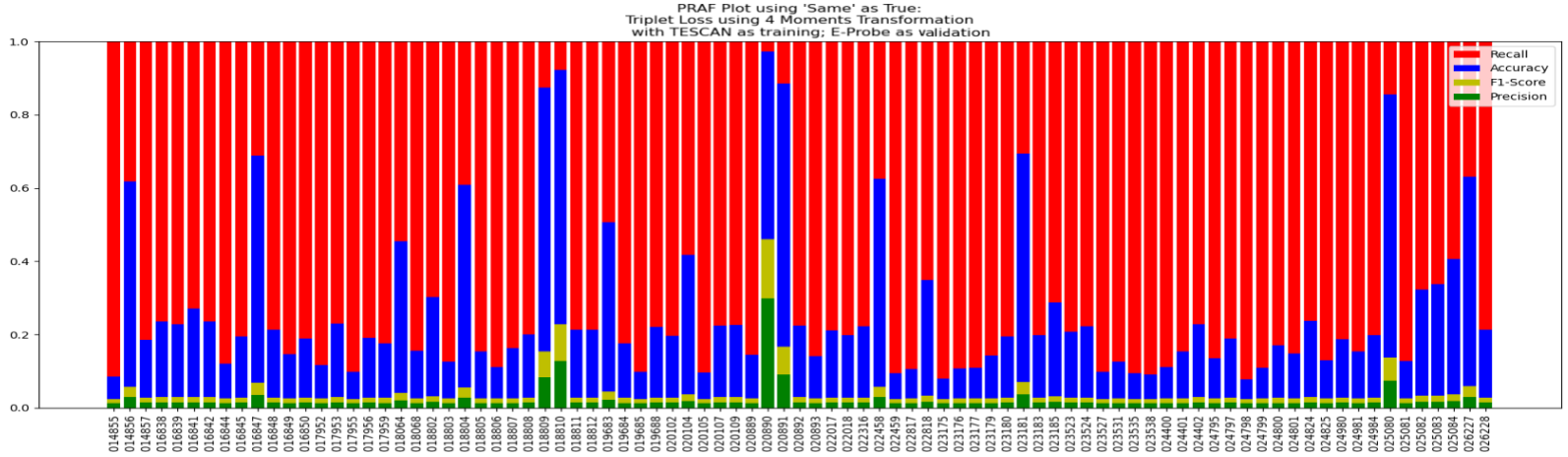

Figure 20. Precision, Recall, Accuracy, and F1-Scores using 'Same' as truth for the model using Triplet Loss with a moment transformation.

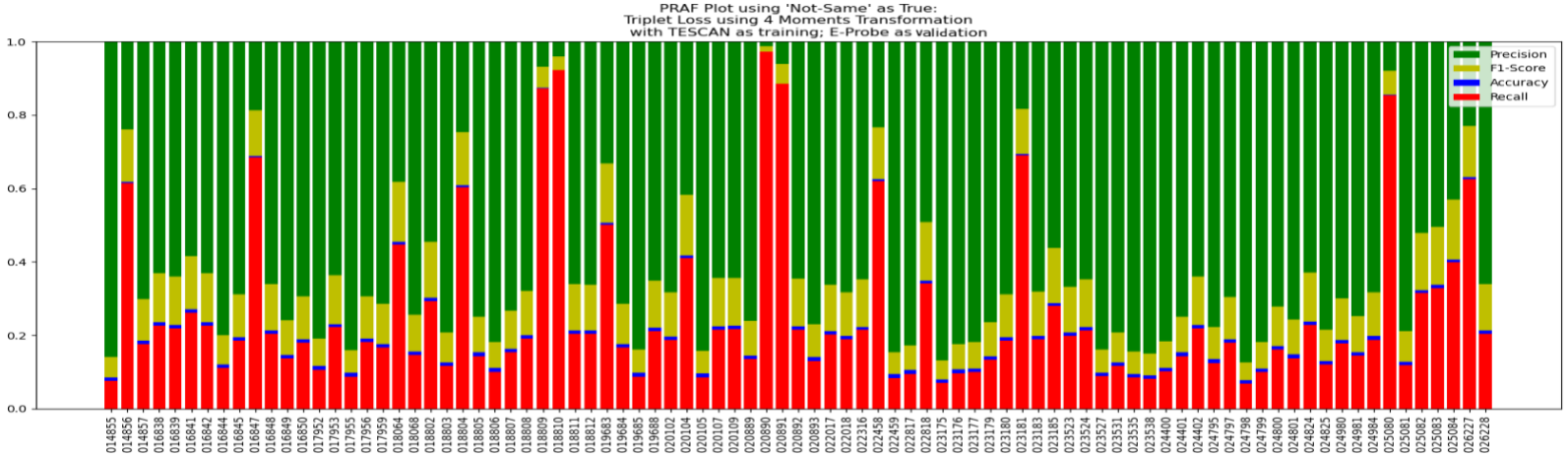

Figure 21. Precision, Recall, Accuracy, and F1-Scores using 'Not-Same' as truth for the model using Triplet Loss with a moment transformation.

Another way to look at these results is that the model can determine when two particle observations represent the same sample, but it is still finding patterns and similarities in particle observations from different samples. However, it is not going to the extreme by assuming all particle observations have similarities. It is important to remember that this model had never seen the EProbe data until it was tasked with making these predictions, which means that training on the TESCAN data alone was enough to produce a model that could distinctly characterize samples. Every other combination (binary cross-entropy with and without moment transformation, and triplet loss without moment transformation) was unable to achieve recall ≈ 1 for predicting 'same' or precision ≈ 1 for predicting 'not-same.'

The final metric to be analyzed is sample similarities. Only a few examples will be shown here. Figure 22 and Table 15 show the similarities the model predicted using triplet loss and moment transformation. This grouping of samples does not correspond entirely to any of the groupings that were provided. It has two samples from group 4, but it is hard to say how the mixed group relates to the other samples. This could imply that the samples within the entire EProbe data set contain a high enough degree of similarity that the model has difficulty accurately labeling samples from a single group as similar. SN020890 is a single sample collected as part of project number 2013-241. Of the other 9 samples found within this project, only SN020891 was labeled as similar. The cause could be one of two things. Either the model is learning erroneous patterns during training, or there are currently unknown, underlying patterns that the model is deciphering. If the prior is true, then adjusting the hyper parameters and improving model performance would reduce the appearance of erroneous patterns. If the latter is true, then further knowledge of the origin of the samples would be needed to assess the accuracy of the predicted groupings.

Table 15. A grouping of samples that were predicted similar to SN020890 using triplet loss and moment transformation.

Sample ID	Original Group	Similarity
SN020890	Mixed	1.000
SN018809	Group 4	0.494
SN018810	Group 4	0.921
SN020891	Mixed	0.515
SN025080	Group 2	0.366
SN026227	Mixed	0.058

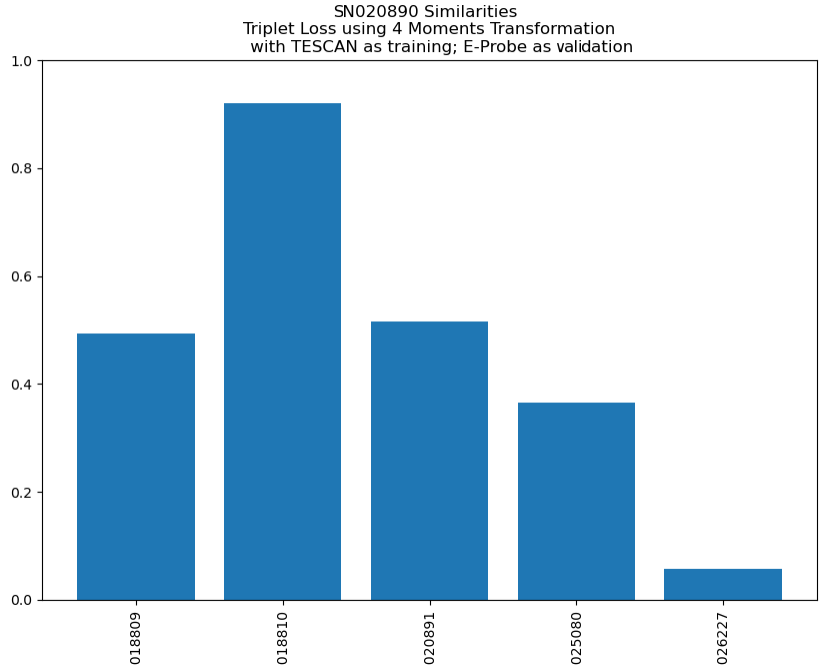

Figure 22. A grouping of samples that were predicted similar to SN020890 using triplet loss and moment transformation.

Another example is of sample SN023181 from group 3. Group 3 contains 12 different samples. However, the model only predicted 2 other samples from group 3 as similar to SN023181, and the similarities were predicted as very low. This is

shown in Figure 23. It is also shown that SN026227 was predicted as most similar to SN023181, but the reason for this is unknown.

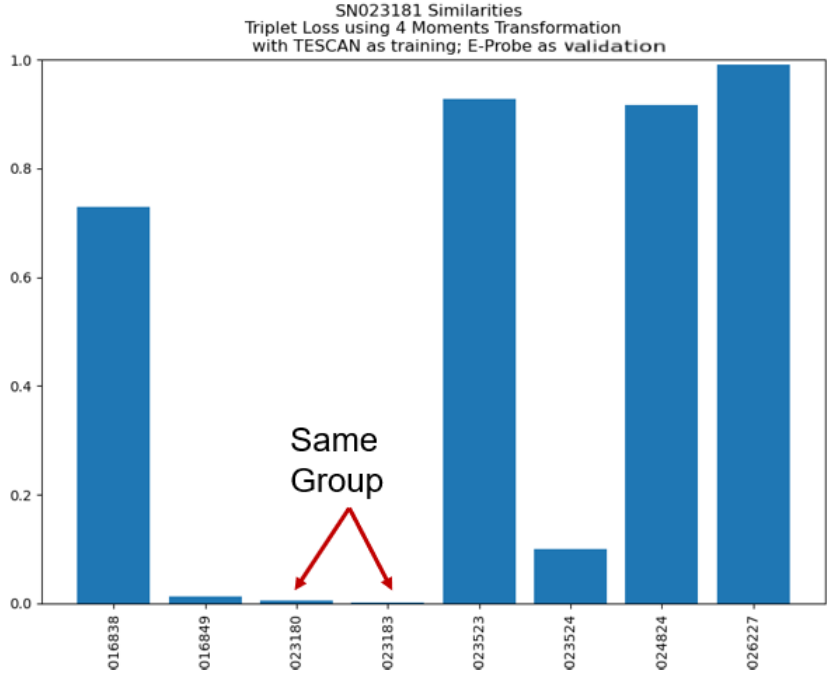

Figure 23. A grouping of samples that were predicted similar to SN023181.

Performing this analysis in the blind, only speculations can be made as to why the model chose these samples as similar. However, a few observations were made when comparing samples SN020890, SN020891, and SN018810 to a different sample, SN022458. This sample was chosen at random out of all the samples that were deemed 'not-similar'. SN022458 had noticeably more iron (7.78% average versus 4.68% maximum), more magnesium (5.12% average versus 1.54% maximum), and less uranium (0.157% average versus 70-80% average). It was also the only sample out of these four to contain any of the following elements: Ba, Bi, Ce, Dy, Hf, La, Nd, Pr, Sc, Sr, Th, Y, Yb, Zr. Taking the ratio of average zirconium to average hafnium gives a value of about 37.4. This falls within the range of ratios for the natural abundances of these two elements. This, along with the low uranium content and diverse range of elements present, imply that SN022458 could contain a sizable

portion of natural soil. Whether it contains only natural soil or is an anthropogenic product is uncertain. Further, this grouping was chosen because it was the simplest grouping. For example, sample SN014856 is from group 1, which contains 3 different samples, and sample SN018804 is from group 4, which contains 12 different samples. Figures 24 and 25 show that the model sometimes over associates samples to one another. The model predicted nearly all the samples present in the EProbe data as similar to both SN014856 and SN018804. Binary cross-entropy and triplet loss with no transformations trended towards having all samples similar to one another. Using the moment transformation appears to have added distinction to the data that the models could more easily register. This trend was seen during all runs and it does not appear that any predicted groupings made by the models coincide with the provided groupings.

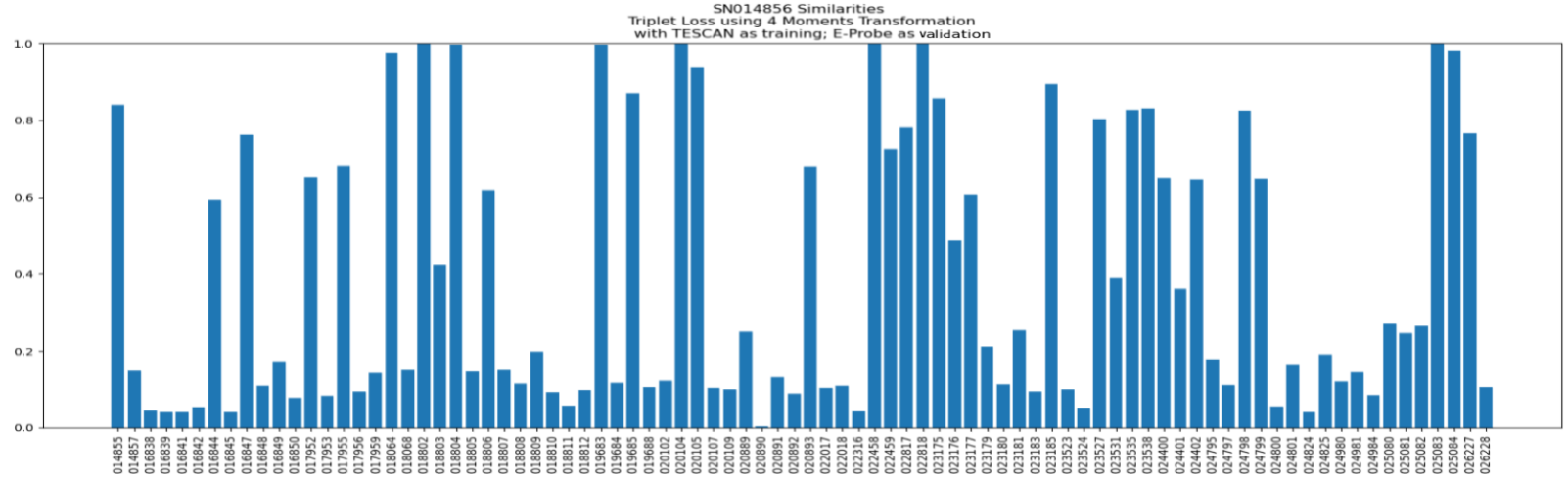

Figure 24. A grouping of samples that were predicted similar to SN014856 using triplet loss and moment transformation.

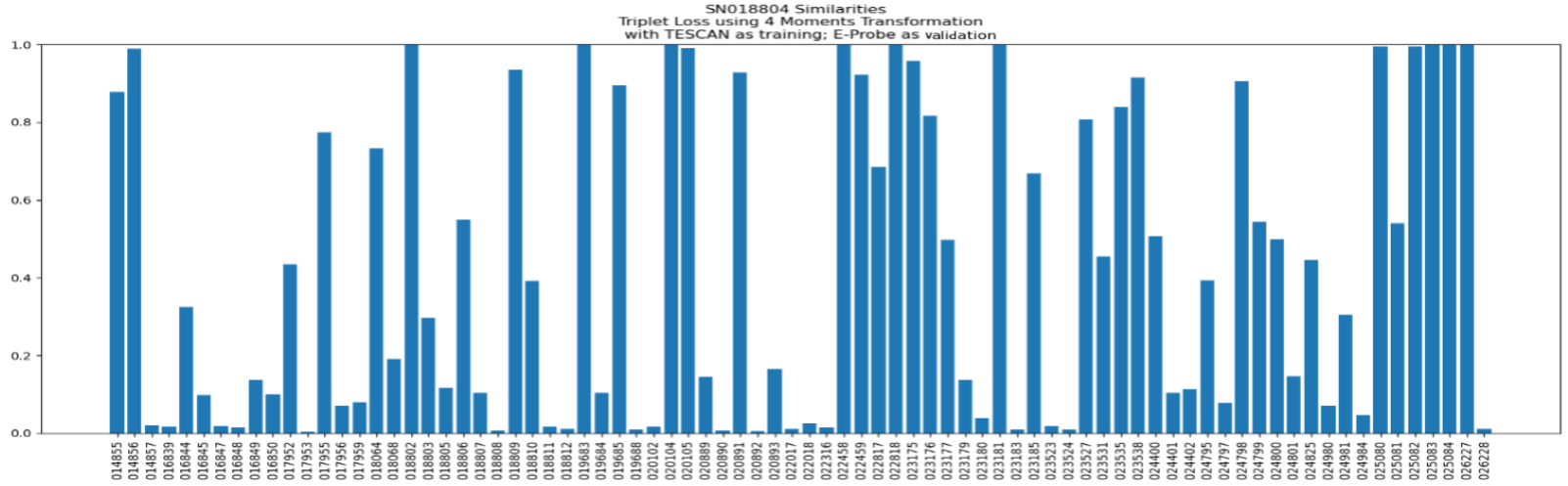

Figure 25. A grouping of samples that were predicted similar to SN018804 using triplet loss and moment transformation.

4.2.2 Trained on EProbe and Validated with TESCAN

Figures 26 through 29 show the accuracies plotted versus the training epoch. These graphs show that a model trained on the EProbe data and using triplet loss has worse performance when compared to binary cross-entropy loss. Again, implementing batch hard or semi-hard could resolve this performance issue. However, these graphs do show that a moment transformation improves the performance of the model, which agrees with the models trained on TESCAN data and validated with EProbe. Binary cross-entropy also appears more stable in this scenario.

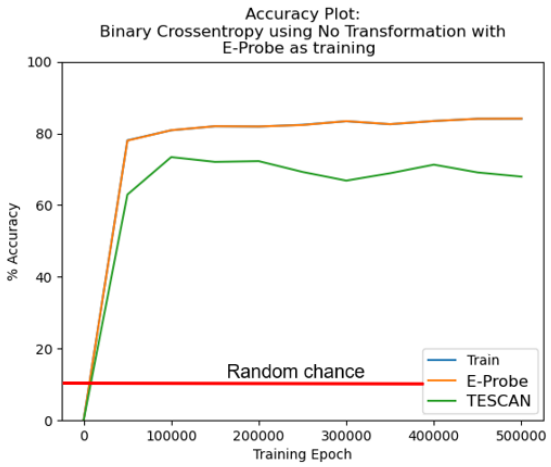

Figure 26. Binary Cross-entropy with no Transformation.

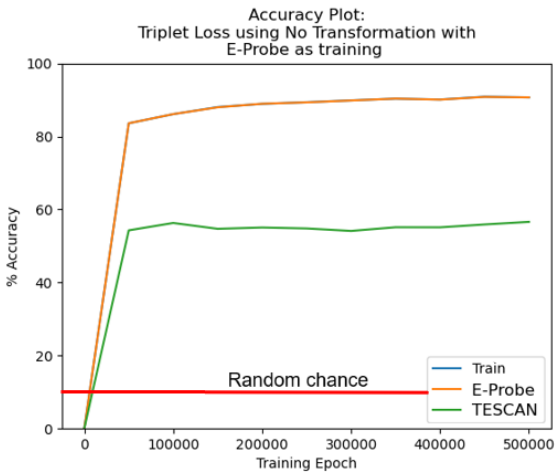

Figure 27. Triplet Loss with no transformation.

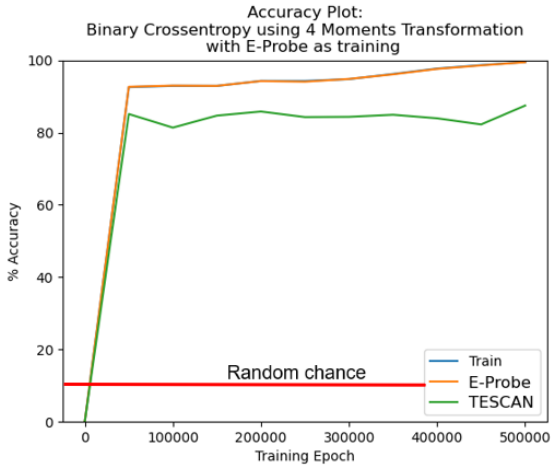

Figure 28. Binary Cross-entropy with 4 moment transformation.

Figure 29. Triplet Loss with 4 moment transformation.

Figures 30 and 31 show the accuracy and loss, respectively, of a different triplet loss and moment transformation run. Unlike before, the loss does not appear to decrease noticeably over the course of training. Instead it appears that the model is constantly trying to find a better solution by decreasing the loss, but is unable to figure out what that solution is. The issue here appears to be that the model is overfitting on the low variance EProbe data, so making predictions on the higher variance TESCAN data is difficult for the model.

Figure 30. Accuracy of a model using Triplet Loss with 4 moment transformation versus training epoch.

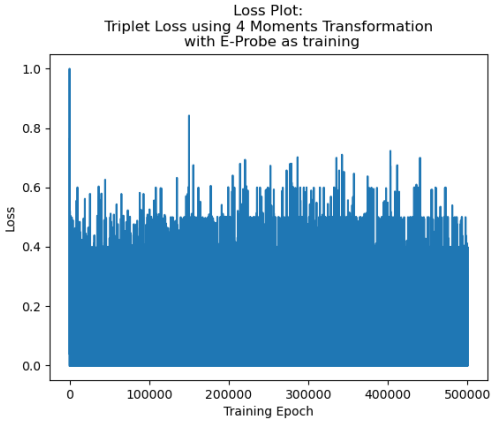

Figure 31. Loss of a model using Triplet Loss with 4 moment transformation versus training epoch.

It is difficult to compare accuracies between this scenario (training on EProbe and validation on TESCAN) and the previous one (training on TESCAN and validation on EProbe). Random chance accuracy on the previous scenario is approximately 1.15%, whereas this scenario has fewer samples during the validation runs (87 versus 10). A random chance accuracy for this scenario is $\frac{1}{10} = 10\%$. Binary cross-entropy with moment transformation had consistently better accuracy during each scenario, so a relative ratio can be obtained by taking the peak accuracy for that model over the random chance accuracy. This scenario gives a ratio of $\frac{87.5}{10} = 8.75$. The previous scenario gives a ratio of approximately $\frac{79}{1.15} \approx 68.7$. This implies that the previous scenario has a better relative accuracy. However, each scenario provides its own strengths and weaknesses, and this ratio is only a single metric that can be used to measure success.

The PRAF scores for triplet loss using the moment transformation are shown in Figures 32 and 33. This model combination is being highlighted for similar reasons as in the previous scenario. Its micro average for recall when predicting 'same' is 0.75, which is the highest amongst the models examined. This means the model more

frequently predicts correctly when it is presented with two particle observations from the same sample. Its micro average for precision when predicting 'not-same' is 0.94, which is again the highest amongst the models examined. As in, the model more frequently makes accurate not-same predictions. These values are distinctly lower than from the previous scenario, which implies that the models have a more difficult time learning from the EProbe data. Binary cross-entropy with moment transformation had a distinctly higher accuracy in this scenario, so it is important to compare that model to triplet loss with moment transformation. The precision and recall micro average when predicting 'not-same' is 0.92 and 0.94, respectively. The precision and recall micro average when predicting 'same' is 0.35 and 0.27, respectively. This model made a total of 92,066 not-same predictions and 7,934 same predictions. This implies that the model was very efficient at predicting not-same, but far less efficient when predicting same. However, during validation the models were forced to make far more not-same predictions, which skewed the accuracies in favor of making correct not-same predictions. The model using binary cross-entropy with triplet loss had higher accuracy simply because it was more likely to make not-same predictions.

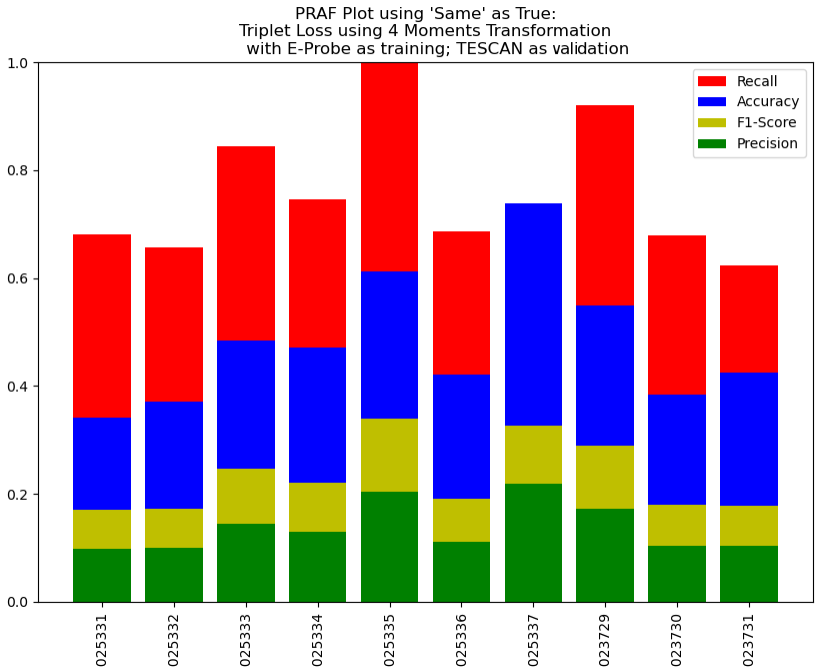

Figure 32. Precision, Recall, Accuracy, and F1-Scores using 'Same' as truth for the model using Triplet Loss with a moment transformation.

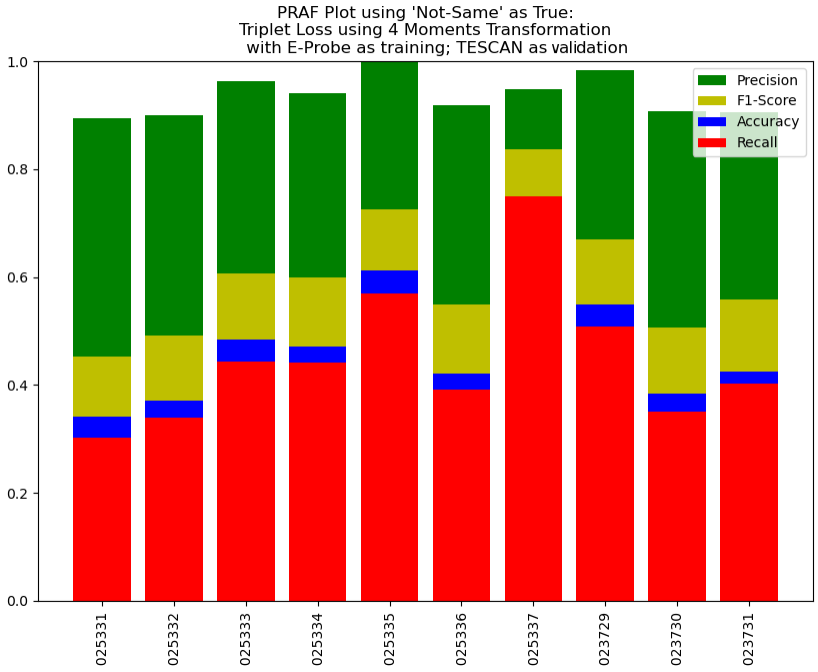

Figure 33. Precision, Recall, Accuracy, and F1-Scores using 'Not-Same' as truth for the model using Triplet Loss with a moment transformation.

It is no surprise that the model using binary cross-entropy with moment transformation did not produce high similarity scores. The two samples the model predicted as most similar were SN025333 and SN025334 with roughly a 50% similarity. Both of these samples originate from Project 2018-204 implying there may be a correlation in their origin. All other samples were predicted at less than 40% similarity to every other sample. With a few exceptions, the general trend was for the model to predict samples from the same project as similar, which is the desired outcome for these models. On the other hand, the model using triplet loss with moment transformation gave a wide range of predicted similarities, and every sample was predicted as similar to each other sample to some degree. Further, there did not appear to be a higher association between samples originating from the same project. There did not appear to be any trend for which samples were predicted as similar. As before, the cause could be one of two things. Either the model is learning erroneous patterns during training, or there are currently unknown, underlying patterns that the model is deciphering. It is difficult to ascertain any underlying patterns without additional information regarding the sample origins. This, along with the large number of particle observations found in the TESCAN data, make it difficult to analyze the quality of these similarity predictions. When training on EProbe data and validating on TESCAN data, binary cross-entropy appears to be stronger at associating accurate similarities. An example of similarities from each model is shown in Figures 34 and 35.

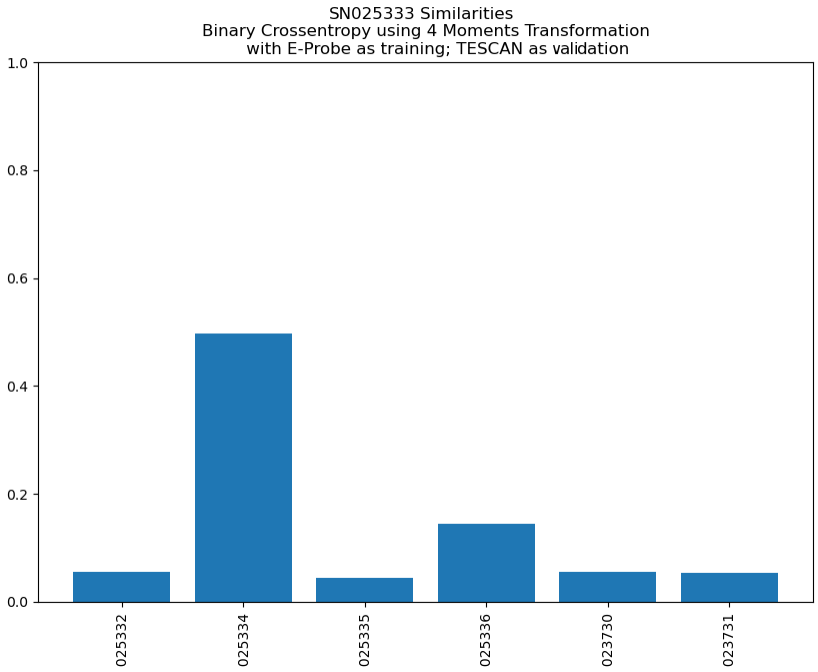

Figure 34. A grouping of samples that were predicted similar to SN025333 using binary cross-entropy and moment transformation.

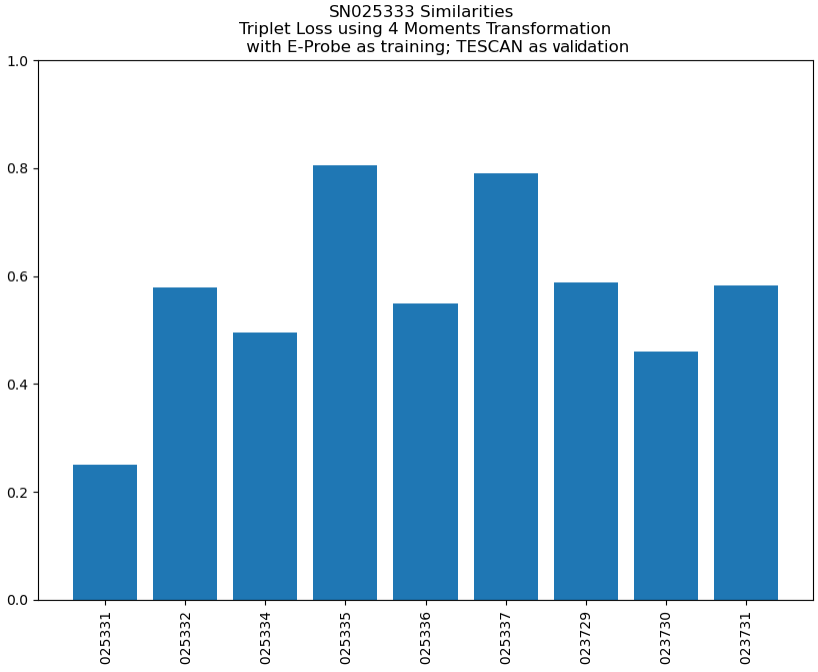

Figure 35. A grouping of samples that were predicted similar to SN025333 using triplet loss and moment transformation.

V. Conclusion

There is a great deal of interest in figuring out a means to quickly relate samples with measured physical properties to one another. A common proposal is to use machine learning techniques to discern patterns that are difficult for humans to distinguish in a reasonable amount of time. For the nuclear forensics community, traditional machine learning techniques, such as linear models and decision tree classifiers, have been used to relate specific particle observations to the samples they belong to. However, this does not solve the issue of figuring out how similar two different samples are to one another. This research proposed using a Siamese network with triplet loss to predict sample similarities based on individual observations. An important step in accomplishing this was to apply a moment transformation to the data. This provided a means to load balance the data and encapsulate the essence of the data into a few meta observations. The Siamese network was then trained separately on two distinct elemental assay datasets that represent what is often seen in nuclear forensics.

5.1 Research Conclusions

There were two important outcomes of a model using moment transformation and trained on the TESCAN data. The first was that almost every time the model was presented with a pair of observations from the same sample it predicted same. The second was that almost every time the model predicted not-same it was a correct prediction. Both of these together indicate that using moment transformation will give reliable predictions of two observations from different samples as being different. As in, if the model gives a not-same prediction, there is assurance that the prediction is correct.

The moment transformation was originally meant to set up the data to be used in a one-shot learning method. A single particle observation, especially in a dataset similar to the TESCAN data, does not always represent the whole sample. However, when a sample is collected there is never a guarantee that a large number of particle observations can be obtained, so this transformation was meant to level the playing field and holistically represent a sample. In all but one scenario presented, using a moment transformation appeared to be beneficial and increased all the PRAF scores of each model. The one exception to this is when the model using binary cross-entropy with moment transformation was trained on the EProbe data. The recall for predicting same was significantly lower when compared to no transformation. This is likely due to the very low variance seen in the EProbe data. The model erroneously learns that samples are very distinct and will never have much in the way of typical background soil concentrations. When this model was validated using the TESCAN data, it was unsure how to handle the large variance seen in that dataset. When validating on TESCAN data, guessing same on every combination would give a precision and recall for predicting same of 0.1 and 1.0, respectively. Based on this, the model using binary cross-entropy predicted a large number of observation pairs from the same sample as different. In contrast, this model did achieve the highest precision for predicting same out of all models and scenarios looked at. This says that the model was more likely to make not-same predictions, but when it did make a same prediction it was much more accurate than the other models. This is the opposite of the goal of this research. It is preferred that a model find patterns and connections between different samples and predict same more often. So if using a model with binary cross-entropy and moment transformation, it is not recommended to train it on EProbe-like data alone.

In contrast, a model using binary cross-entropy with no transformation is only

marginally better than random chance. When trained on the EProbe data, this model trended towards an increased recall for not-same predictions, but this could be a predisposition for predicting not-same. When trained on the TESCAN data, the model predicted same much more often, however it does not appear that those predictions were any better than random. It would not be recommended to use a model with binary cross-entropy with no transformation unless a lot of work could be done to the network architecture to improve the PRAF scores.

Despite these drawback with binary cross-entropy, a model using triplet loss showed a great deal of potential. As mentioned, a model that predicts same more often would be more likely to find the underlying patterns amongst the samples, which is how the triplet loss models behaved. The models using triplet loss achieved much higher recall scores for predicting same and higher precision scores for predicting not-same, when compared to binary cross-entropy. However, in its current state, the models using triplet loss are not free of issues as they were somewhat volatile. It appeared that local minima were hit when performing the gradient descent during training. With no transformation, the precision for predicting both same and not-same was higher than random chance, the recall for predicting same was significantly greater than random chance, and the recall for predicting not-same at or below random chance. All this says that triplet loss by itself is not necessarily a strong predictor. When the moment transformation was used, all the PRAF scores increased to a point noticeably greater than random chance. It did trend towards a lower accuracy compared to the model with binary cross-entropy using moment transformation, but again, binary cross-entropy using moment transformation trended towards not-same prediction. In short, the models using triplet loss still need some optimization before they can achieve higher PRAF scores. However, they were much more versatile in what data they could be trained on as they performed more consistently across the

two datasets.

When comparing the Siamese networks to the traditional machine learning techniques, it is difficult to directly compare the success metrics. The traditional machine learning techniques used are classifiers which attempt to place an observation into a known category, whereas the Siamese network simply predicts how similar one observation is to another. The Siamese network had generally higher macro average scores for both datasets. The exception is the precision for predicting same. However, this does not specifically indicate a worse performing model. The Siamese network is not attempting to categorize the observations, it is simply giving a similarity score, which is the true strength of the Siamese network. The traditional machine learning techniques explored require prior knowledge of the categories that the training samples belong to, which is not true for the Siamese network. The prediction it makes comes from how the similarity score relates to the threshold and margin hyper parameters. Optimizing these hyper parameters would likely increase the performance of the Siamese network.

5.2 Recommendations for Application

The model using binary cross-entropy made more meaningful connections when trained on the EProbe data. Conversely, the model using triplet loss made more meaningful connections when trained on the TESCAN data. These two datasets can be thought of as opposite extremes, and any database will likely contain a spectrum of such datasets. As such, it would be beneficial to use both loss functions during real world applications until improvements could be made to the triplet loss function. Additionally, setting aside a test set from the known samples is crucial in ensuring the trained model is working as intended, and should be done before any unknown samples are fed through the network.

When putting these models into practice, there needs to be a mindful consideration of what type of data is being fed through the models. The models were trained using elemental assay data, and the samples collected are suspected to contain at least some trace remnant of a byproduct originating from some part of the nuclear fuel cycle. Deviation from this type of sample, to include isotopic assays, could lead to a trained model that outputs little to no meaningful results. Additional research would need to be conducted to verify a model trained using alternate types of data. However, staying within the realm of elemental assays, a model using these techniques could prove very useful in determining sample origins. Using moment transformation, these samples can be analyzed more holistically and compared to one another at a gross level, and any unknown sample that is deemed similar to a known sample from a database will be more likely to share characteristics with that known database sample. In its current state, it is still important to review any prediction made by the models, but these techniques could reduce analysis time and effort by directing focus to more meaningful samples.

This leads into the types of data that should be used during training. The TESCAN data appears to contain more than just particles of interest. There are environmental signatures that increase the variability within a single sample, which gives the model a more diverse sense of what makes the sample unique. This is a likely contributor to the higher PRAF scores seen when training on the TESCAN data. However, datasets similar to the EProbe data are more common and will likely be the main contributing factor when training a model. Because of this, it may be prudent to provide the model samples that are known to contain nothing of interest during training. This could give an indication of which samples contain some amount of environmental signatures, or potentially indicate which unknown samples contain nothing of interest. Additionally, it could prove useful to train the model using a di-

verse set of distinct samples. The more sample origins a model can distinctly identify during training, the better equipped the model will be at accurately attributing an unknown sample to a known sample.

5.3 Future Work

One intent of this research was to implement a triplet loss using batch hard or semi-hard, however this was not completed. It is recommended that a correct implementation of batch hard be researched. This could overcome the local minima that the triplet loss network appears to be getting stuck in, and it could help the model converge much sooner. This would reduce the mount of time required for training the network while still providing accurate results.

Using a moment transformation generally showed improvements to the prediction capabilities. However, this was a placeholder for a potentially more capable transformation technique. There is the option for using a different number of moments, and how many moments are required to give good results. Looking more broadly, a set invariant transformation is not limited to using moments. Looking into set invariance and finding a transformation technique that is especially effective with the data found in this research could prove very useful.

It was shown that the type of dataset that was used to train the network could drastically influence the prediction capabilities of the network. Experimenting with the type of dataset and the samples found within is important in order to understand the limitations of any machine learning technique. It is nice to imagine that there is a one-size-fits-all solution, but practically speaking it is not realistic to expect one tool to do every job. More research into how the datasets influence the network and how to overcome those hurdles is recommended.

Much of this research was meant as a proof of concept. As such the Siamese

network used, and all the relating hyper parameters, have a great deal of room for optimization. The threshold and margin hyper parameters were not optimized much and could be tuned alongside a batch hard process to increase the capabilities of using triplet loss. The network itself is simple, and in the grand scheme of deep learning, not that deep. With many corporate ANNs being hundreds of layers deep, it should not be expected for a three layer deep network to adequately compete. There are countless possible network combinations, so there is plenty of room for this network to be improved.

Appendix A. Data Summary

Table 16. Project TESCAN data summary.
***This represents the total number of unique elemental features and each project will have overlapping features with other projects.**

Project ID	Sample ID	Particle Observations	Elemental Features
	SN023729	601	36
2017-162	SN023730	15,191	36
	SN023731	4,429	36
Subtotal	3 Samples	20,221	36
	SN025331	2,538	39
	SN025332	19,013	39
	SN025333	2,620	39
2018-204	SN025334	391	39
	SN025335	1,262	39
	SN025336	13,605	39
	SN025337	18,814	39
Subtotal	7 Samples	58,243	39
Total	10 Samples	78,464	40*

Table 17. Project EProbe data summary.
***This represents the total number of unique elemental features and each project will have overlapping features with other projects.**

Group	**Project ID**	**Sample ID**	Particle Observations	Elemental Features
g1	06-111	SN014855	16	14
		SN014856	19	17
		SN014857	2	12
g4	11-031	SN018800	8	1
		SN018802	17	23
		SN018803	18	20
		SN018804	15	15
		SN018805	19	20
		SN018806	16	20
		SN018807	22	20
		SN018808	20	16
		SN018809	16	22
		SN018810	17	18
		SN018811	18	20
		SN018812	19	19
g5	15-066	SN022017	28	16
		SN022018	16	15

Continuation of Table 17				
Group	**Project ID**	**Sample ID**	Particle Observations	Elemental Features
g2	08-201	SN016837	12	2
		SN016838	25	17
		SN016839	22	23
		SN016840	12	2
		SN016841	26	24
		SN016842	20	23
		SN016843	24	4
		SN016844	25	17
		SN016845	20	35
		SN016846	19	8
		SN016847	28	20
		SN016848	19	25
		SN016849	27	21
		SN016850	12	11
	12-089	SN019688	20	28
	18-167	SN024980	15	10
		SN024981	16	10
	10-068	SN017959	8	15
	10-065	SN017952	19	18
		SN017953	23	18
	12-086	SN019683	16	19
		SN019684	14	18
		SN019685	21	12

Continuation of Table 17				
Group	**Project ID**	**Sample ID**	Particle Observations	Elemental Features
g2	18-167	SN024980	15	10
		SN024981	16	10
	18-227	SN024984	8	10
	18-246	SN025079	19	9
		SN025080	18	34
		SN025081	23	27
		SN025082	16	20
		SN025083	18	28
		SN025084	22	25

Continuation of Table 17				
Group	**Project ID**	**Sample ID**	Particle Observations	Elemental Features
g3	15-210	SN022456	11	1
		SN022457	17	3
		SN022458	28	12
		SN022459	29	22
		SN022460	26	5
		SN022464	11	4
	16-083	SN022817	23	20
		SN022818	7	10
		SN022819	10	1
		SN022823	10	5
	16-224	SN023170	26	5
		SN023172	10	1
		SN023174	12	1
		SN023175	32	15
		SN023176	35	20
		SN023177	27	15
		SN023178	26	4
		SN023179	28	13
		SN023180	26	17
		SN023181	15	18
		SN023182	29	5
		SN023183	11	20
		SN023185	30	15
		SN023186	28	6

Continuation of Table 17				
Group	**Project ID**	**Sample ID**	Particle Observations	Elemental Features
mixed	18-158	SN024822	11	1
		SN024823	15	6
		SN024824	31	21
		SN024825	25	12
	19-312	SN026227	37	23
		SN026228	18	20
	12-216	SN020087	18	3
		SN020088	13	1
		SN020089	18	2
		SN020090	23	7
		SN020091	17	3
		SN020092	15	5
		SN020093	20	4
	12-221	SN020100	15	2
		SN020101	15	2
		SN020102	13	22
		SN020104	19	10
		SN020105	10	12
		SN020107	26	18
		SN020109	20	13

Continuation of Table 17				
Group	**Project ID**	**Sample ID**	Particle Observations	Elemental Features
mixed	13-241	SN020885	8	1
		SN020886	18	3
		SN020887	18	4
		SN020888	13	1
		SN020889	14	16
		SN020890	12	24
		SN020891	14	20
		SN020892	23	18
		SN020893	21	14
	15-174	SN022312	10	2
		SN022313	13	6
		SN022314	23	5
		SN022315	25	6
		SN022316	27	20
	17-049	SN023522	15	3
		SN023523	20	22
		SN023524	17	20
		SN023527	28	10
		SN023528	14	2
		SN023531	27	20
		SN023535	22	11
		SN023538	22	14

Continuation of Table 17				
Group	**Project ID**	**Sample ID**	Particle Observations	Elemental Features
mixed	18-131	SN024794	24	9
		SN024795	29	14
		SN024796	24	3
		SN024797	35	20
		SN024798	30	10
		SN024799	32	18
		SN024800	30	20
		SN024801	31	13
	18-034	SN024399	17	1
		SN024400	31	12
		SN024401	34	20
		SN024402	30	16
	10-088	SN018062	14	2
		SN018064	22	17
		SN018065	11	9
		SN018066	14	3
		SN018067	14	2
		SN018068	22	12
		SN018069	22	9
		SN018070	18	6
		SN018071	10	2
		SN018072	24	9
End of Table				

Appendix B. Traditional Machine Learning PRAF Scores

Table 18. Precision, Recall, Accuracy, and F1-Scores for Logistic Regression trained on EProbe Data.

Sample ID	Precision	Recall	F1-Score	Support
SN014855	1.000	0.800	0.889	5
SN014856	0.667	0.500	0.571	8
SN014857	0.222	1.000	0.364	4
SN016837	0.000	0.000	0.000	1
SN016838	0.333	0.250	0.286	8
SN016839	0.375	0.429	0.400	7
SN016840	0.000	0.000	0.000	0
SN016841	0.000	0.000	0.000	9
SN016842	0.125	0.154	0.138	13
SN016843	0.000	0.000	0.000	1
SN016844	0.000	0.000	0.000	9
SN016845	0.250	0.154	0.190	13
SN016846	0.000	0.000	0.000	2
SN016847	0.000	0.000	0.000	10
SN016848	0.250	0.222	0.235	9
SN016849	0.167	0.143	0.154	7
SN016850	0.091	0.333	0.143	3
SN017952	0.750	0.300	0.429	10
SN017953	0.000	0.000	0.000	7
SN017955	0.500	0.500	0.500	6
SN017956	0.267	0.667	0.381	6
SN017959	1.000	0.444	0.615	9
SN018062	0.000	0.000	0.000	1
SN018064	0.000	0.000	0.000	11
SN018065	0.500	0.500	0.500	4
SN018066	0.000	0.000	0.000	1
SN018068	0.500	0.143	0.222	7
SN018069	0.077	0.250	0.118	4
SN018070	0.000	0.000	0.000	3
SN018071	0.000	0.000	0.000	1
SN018072	0.000	0.000	0.000	2

Continuation of Table 18				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN018800	0.000	0.000	0.000	1
SN018802	0.667	0.182	0.286	11
SN018803	0.750	0.643	0.692	14
SN018804	0.333	0.333	0.333	3
SN018805	0.111	0.111	0.111	9
SN018806	0.500	0.286	0.364	7
SN018807	0.600	0.429	0.500	7
SN018808	0.000	0.000	0.000	4
SN018809	0.600	0.429	0.500	7
SN018810	0.667	0.333	0.444	6
SN018811	0.800	0.571	0.667	7
SN018812	0.250	0.286	0.267	7
SN019683	0.375	0.375	0.375	8
SN019684	0.000	0.000	0.000	9
SN019685	0.000	0.000	0.000	7
SN019688	0.400	0.118	0.182	17
SN020087	0.000	0.000	0.000	0
SN020088	0.000	0.000	0.000	0
SN020089	0.000	0.000	0.000	1
SN020090	0.000	0.000	0.000	3
SN020091	0.000	0.000	0.000	2
SN020092	0.000	0.000	0.000	3
SN020093	0.000	0.000	0.000	1
SN020100	0.000	0.000	0.000	2
SN020101	0.000	0.000	0.000	1
SN020102	0.143	0.111	0.125	9
SN020104	0.000	0.000	0.000	3
SN020105	0.167	0.250	0.200	4
SN020107	0.250	0.250	0.250	4
SN020109	0.000	0.000	0.000	5
SN020885	0.000	0.000	0.000	0
SN020886	0.000	0.000	0.000	1
SN020887	0.000	0.000	0.000	2
SN020888	0.000	0.000	0.000	1

Continuation of Table 18				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN020889	0.000	0.000	0.000	2
SN020890	0.286	0.167	0.211	12
SN020891	0.300	0.667	0.414	9
SN020892	0.000	0.000	0.000	6
SN020893	0.200	0.200	0.200	5
SN022017	0.200	0.250	0.222	4
SN022018	0.286	0.222	0.250	9
SN022312	0.000	0.000	0.000	0
SN022313	0.200	0.500	0.286	2
SN022314	0.000	0.000	0.000	1
SN022315	0.000	0.000	0.000	3
SN022316	0.000	0.000	0.000	9
SN022457	0.000	0.000	0.000	0
SN022458	0.000	0.000	0.000	4
SN022459	0.400	0.250	0.308	8
SN022460	0.000	0.000	0.000	2
SN022464	0.333	0.500	0.400	2
SN022817	0.000	0.000	0.000	4
SN022818	0.172	1.000	0.294	5
SN022819	0.000	0.000	0.000	1
SN022823	0.200	0.333	0.250	3
SN023170	0.000	0.000	0.000	3
SN023172	0.000	0.000	0.000	1
SN023175	0.000	0.000	0.000	6
SN023176	0.400	0.200	0.267	10
SN023177	0.250	0.286	0.267	7
SN023178	0.000	0.000	0.000	2
SN023179	0.000	0.000	0.000	3
SN023180	0.000	0.000	0.000	6
SN023181	0.750	0.500	0.600	6
SN023182	0.000	0.000	0.000	4
SN023183	0.833	0.833	0.833	6
SN023185	0.125	0.200	0.154	5
SN023186	0.000	0.000	0.000	4

Continuation of Table 18				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN023522	0.000	0.000	0.000	0
SN023523	0.333	0.167	0.222	6
SN023524	0.000	0.000	0.000	11
SN023527	0.000	0.000	0.000	5
SN023528	0.000	0.000	0.000	0
SN023531	0.000	0.000	0.000	8
SN023535	0.000	0.000	0.000	5
SN023538	0.000	0.000	0.000	2
SN024399	0.000	0.000	0.000	1
SN024400	0.000	0.000	0.000	4
SN024401	0.000	0.000	0.000	5
SN024402	0.200	0.143	0.167	7
SN024794	0.000	0.000	0.000	4
SN024795	0.250	0.250	0.250	4
SN024796	0.000	0.000	0.000	1
SN024797	0.250	0.125	0.167	8
SN024798	0.000	0.000	0.000	2
SN024799	0.000	0.000	0.000	9
SN024800	0.000	0.000	0.000	7
SN024801	0.500	0.250	0.333	4
SN024822	0.000	0.000	0.000	1
SN024823	0.000	0.000	0.000	3
SN024824	0.222	0.222	0.222	9
SN024825	0.000	0.000	0.000	3
SN024980	0.333	1.000	0.500	2
SN024981	0.500	0.167	0.250	6
SN024984	0.714	0.833	0.769	6
SN025079	0.333	0.333	0.333	3
SN025080	0.846	0.733	0.786	15
SN025081	0.571	0.444	0.500	9
SN025082	0.000	0.000	0.000	7
SN025083	0.500	0.400	0.444	15
SN025084	0.200	0.250	0.222	8
SN026227	0.500	0.455	0.476	11

Continuation of Table 18				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN026228	0.778	0.875	0.824	8
accuracy			0.237	700
macro avg	0.191	0.181	0.171	700
weighted avg	0.283	0.237	0.241	700
End of Table				

Table 19. Precision, Recall, Accuracy, and F1-Scores for Linear Discriminate Analysis trained on EProbe Data.

Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN014855	0.800	0.800	0.800	5
SN014856	0.800	0.500	0.615	8
SN014857	0.200	1.000	0.333	4
SN016837	0.000	0.000	0.000	1
SN016838	0.000	0.000	0.000	8
SN016839	0.167	0.143	0.154	7
SN016841	0.000	0.000	0.000	9
SN016842	0.125	0.154	0.138	13
SN016843	0.000	0.000	0.000	1
SN016844	0.000	0.000	0.000	9
SN016845	0.200	0.231	0.214	13
SN016846	0.111	0.500	0.182	2
SN016847	0.000	0.000	0.000	10
SN016848	0.100	0.111	0.105	9
SN016849	0.500	0.143	0.222	7
SN016850	0.000	0.000	0.000	3
SN017952	0.000	0.000	0.000	10
SN017953	0.000	0.000	0.000	7
SN017955	0.200	0.500	0.286	6
SN017956	0.286	0.333	0.308	6
SN017959	0.571	0.444	0.500	9
SN018062	0.000	0.000	0.000	1
SN018064	0.000	0.000	0.000	11

Continuation of Table 19				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN018065	0.375	0.750	0.500	4
SN018066	0.000	0.000	0.000	1
SN018068	0.000	0.000	0.000	7
SN018069	0.040	0.250	0.069	4
SN018070	0.000	0.000	0.000	3
SN018071	0.000	0.000	0.000	1
SN018072	0.000	0.000	0.000	2
SN018800	0.000	0.000	0.000	1
SN018802	0.000	0.000	0.000	11
SN018803	0.000	0.000	0.000	14
SN018804	0.000	0.000	0.000	3
SN018805	0.000	0.000	0.000	9
SN018806	0.000	0.000	0.000	7
SN018807	0.500	0.143	0.222	7
SN018808	0.000	0.000	0.000	4
SN018809	0.143	0.143	0.143	7
SN018810	0.200	0.167	0.182	6
SN018811	0.667	0.571	0.615	7
SN018812	0.000	0.000	0.000	7
SN019683	0.000	0.000	0.000	8
SN019684	0.200	0.222	0.211	9
SN019685	0.000	0.000	0.000	7
SN019688	0.000	0.000	0.000	17
SN020087	0.000	0.000	0.000	0
SN020088	0.000	0.000	0.000	0
SN020089	0.000	0.000	0.000	1
SN020090	0.000	0.000	0.000	3
SN020091	0.000	0.000	0.000	2
SN020092	0.000	0.000	0.000	3
SN020093	0.000	0.000	0.000	1
SN020100	0.000	0.000	0.000	2
SN020101	0.000	0.000	0.000	1
SN020102	0.000	0.000	0.000	9
SN020104	0.000	0.000	0.000	3

Continuation of Table 19				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN020105	0.000	0.000	0.000	4
SN020107	0.500	0.250	0.333	4
SN020109	0.000	0.000	0.000	5
SN020886	0.000	0.000	0.000	1
SN020887	0.000	0.000	0.000	2
SN020888	0.000	0.000	0.000	1
SN020889	0.000	0.000	0.000	2
SN020890	0.154	1.000	0.267	12
SN020891	0.000	0.000	0.000	9
SN020892	0.000	0.000	0.000	6
SN020893	0.071	0.200	0.105	5
SN022017	0.000	0.000	0.000	4
SN022018	0.000	0.000	0.000	9
SN022312	0.000	0.000	0.000	0
SN022313	0.067	0.500	0.118	2
SN022314	0.000	0.000	0.000	1
SN022315	0.000	0.000	0.000	3
SN022316	0.250	0.111	0.154	9
SN022456	0.000	0.000	0.000	0
SN022457	0.000	0.000	0.000	0
SN022458	0.000	0.000	0.000	4
SN022459	0.500	0.125	0.200	8
SN022460	0.000	0.000	0.000	2
SN022464	0.083	1.000	0.154	2
SN022817	0.000	0.000	0.000	4
SN022818	0.000	0.000	0.000	5
SN022819	0.000	0.000	0.000	1
SN022823	0.091	0.333	0.143	3
SN023170	0.000	0.000	0.000	3
SN023172	0.000	0.000	0.000	1
SN023175	0.000	0.000	0.000	6
SN023176	1.000	0.100	0.182	10
SN023177	0.500	0.143	0.222	7
SN023178	0.000	0.000	0.000	2

Continuation of Table 19				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN023179	0.000	0.000	0.000	3
SN023180	0.000	0.000	0.000	6
SN023181	1.000	0.333	0.500	6
SN023182	0.000	0.000	0.000	4
SN023183	0.800	0.667	0.727	6
SN023185	0.125	0.200	0.154	5
SN023186	0.000	0.000	0.000	4
SN023523	0.000	0.000	0.000	6
SN023524	0.267	0.364	0.308	11
SN023527	0.000	0.000	0.000	5
SN023528	0.000	0.000	0.000	0
SN023531	0.000	0.000	0.000	8
SN023535	0.000	0.000	0.000	5
SN023538	0.000	0.000	0.000	2
SN024399	0.000	0.000	0.000	1
SN024400	0.037	0.250	0.065	4
SN024401	0.000	0.000	0.000	5
SN024402	0.000	0.000	0.000	7
SN024794	0.000	0.000	0.000	4
SN024795	0.000	0.000	0.000	4
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	8
SN024798	0.000	0.000	0.000	2
SN024799	0.000	0.000	0.000	9
SN024800	0.333	0.429	0.375	7
SN024801	0.000	0.000	0.000	4
SN024822	0.000	0.000	0.000	1
SN024823	0.333	0.333	0.333	3
SN024824	0.000	0.000	0.000	9
SN024825	0.000	0.000	0.000	3
SN024980	0.000	0.000	0.000	2
SN024981	0.167	0.333	0.222	6
SN024984	0.667	0.333	0.444	6
SN025079	0.000	0.000	0.000	3

Continuation of Table 19				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN025080	0.500	0.400	0.444	15
SN025081	0.300	0.333	0.316	9
SN025082	0.000	0.000	0.000	7
SN025083	0.333	0.200	0.250	15
SN025084	0.600	0.375	0.462	8
SN026227	0.333	0.091	0.143	11
SN026228	0.750	0.375	0.500	8
accuracy			0.149	700
macro avg	0.121	0.120	0.098	700
weighted avg	0.172	0.149	0.133	700
End of Table				

Table 20. Precision, Recall, Accuracy, and F1-Scores for Decision Tree Classifier trained on EProbe Data.

Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN014855	0.500	0.600	0.545	5
SN014856	0.714	0.625	0.667	8
SN014857	0.400	1.000	0.571	4
SN016837	0.000	0.000	0.000	1
SN016838	0.000	0.000	0.000	8
SN016839	0.000	0.000	0.000	7
SN016840	0.000	0.000	0.000	0
SN016841	0.000	0.000	0.000	9
SN016842	0.250	0.154	0.190	13
SN016843	0.000	0.000	0.000	1
SN016844	0.200	0.111	0.143	9
SN016845	0.071	0.077	0.074	13
SN016846	0.000	0.000	0.000	2
SN016847	0.182	0.200	0.190	10
SN016848	0.000	0.000	0.000	9
SN016849	0.182	0.286	0.222	7
SN016850	0.111	0.333	0.167	3

Continuation of Table 20				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN017952	0.000	0.000	0.000	10
SN017953	0.000	0.000	0.000	7
SN017955	0.222	0.333	0.267	6
SN017956	0.250	0.167	0.200	6
SN017959	1.000	0.444	0.615	9
SN018062	0.000	0.000	0.000	1
SN018064	0.200	0.091	0.125	11
SN018065	0.400	0.500	0.444	4
SN018066	0.000	0.000	0.000	1
SN018067	0.000	0.000	0.000	0
SN018068	0.143	0.143	0.143	7
SN018069	0.000	0.000	0.000	4
SN018070	0.000	0.000	0.000	3
SN018071	0.000	0.000	0.000	1
SN018072	0.200	0.500	0.286	2
SN018800	0.000	0.000	0.000	1
SN018802	0.250	0.091	0.133	11
SN018803	0.000	0.000	0.000	14
SN018804	0.000	0.000	0.000	3
SN018805	0.000	0.000	0.000	9
SN018806	0.500	0.571	0.533	7
SN018807	0.250	0.571	0.348	7
SN018808	0.000	0.000	0.000	4
SN018809	0.200	0.429	0.273	7
SN018810	0.000	0.000	0.000	6
SN018811	0.833	0.714	0.769	7
SN018812	0.286	0.286	0.286	7
SN019683	0.333	0.125	0.182	8
SN019684	0.111	0.111	0.111	9
SN019685	0.000	0.000	0.000	7
SN019688	0.000	0.000	0.000	17
SN020087	0.000	0.000	0.000	0
SN020088	0.000	0.000	0.000	0
SN020089	0.000	0.000	0.000	1

Continuation of Table 20				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN020090	0.000	0.000	0.000	3
SN020091	0.000	0.000	0.000	2
SN020092	0.000	0.000	0.000	3
SN020093	0.000	0.000	0.000	1
SN020100	0.000	0.000	0.000	2
SN020101	0.000	0.000	0.000	1
SN020102	0.333	0.222	0.267	9
SN020104	0.250	0.333	0.286	3
SN020105	0.000	0.000	0.000	4
SN020107	0.000	0.000	0.000	4
SN020109	0.000	0.000	0.000	5
SN020885	0.000	0.000	0.000	0
SN020886	0.000	0.000	0.000	1
SN020887	0.000	0.000	0.000	2
SN020888	0.000	0.000	0.000	1
SN020889	0.000	0.000	0.000	2
SN020890	0.357	0.417	0.385	12
SN020891	0.000	0.000	0.000	9
SN020892	0.250	0.167	0.200	6
SN020893	0.125	0.200	0.154	5
SN022017	0.143	0.250	0.182	4
SN022018	0.667	0.222	0.333	9
SN022312	0.000	0.000	0.000	0
SN022313	0.000	0.000	0.000	2
SN022314	0.000	0.000	0.000	1
SN022315	0.000	0.000	0.000	3
SN022316	0.167	0.222	0.190	9
SN022457	0.000	0.000	0.000	0
SN022458	0.000	0.000	0.000	4
SN022459	0.222	0.250	0.235	8
SN022460	0.000	0.000	0.000	2
SN022464	0.000	0.000	0.000	2
SN022817	0.333	0.500	0.400	4
SN022818	0.286	0.800	0.421	5

Continuation of Table 20				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN022819	0.000	0.000	0.000	1
SN022823	0.000	0.000	0.000	3
SN023170	0.000	0.000	0.000	3
SN023172	0.000	0.000	0.000	1
SN023175	0.000	0.000	0.000	6
SN023176	0.222	0.200	0.211	10
SN023177	0.250	0.143	0.182	7
SN023178	0.000	0.000	0.000	2
SN023179	0.300	1.000	0.462	3
SN023180	0.200	0.167	0.182	6
SN023181	0.375	0.500	0.429	6
SN023182	0.000	0.000	0.000	4
SN023183	0.500	1.000	0.667	6
SN023185	0.000	0.000	0.000	5
SN023186	0.000	0.000	0.000	4
SN023522	0.000	0.000	0.000	0
SN023523	0.111	0.167	0.133	6
SN023524	0.167	0.091	0.118	11
SN023527	0.000	0.000	0.000	5
SN023528	0.000	0.000	0.000	0
SN023531	0.143	0.125	0.133	8
SN023535	0.000	0.000	0.000	5
SN023538	0.125	0.500	0.200	2
SN024399	0.000	0.000	0.000	1
SN024400	0.000	0.000	0.000	4
SN024401	0.000	0.000	0.000	5
SN024402	0.000	0.000	0.000	7
SN024794	0.000	0.000	0.000	4
SN024795	0.000	0.000	0.000	4
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	8
SN024798	0.000	0.000	0.000	2
SN024799	0.333	0.111	0.167	9
SN024800	0.222	0.286	0.250	7

Continuation of Table 20				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN024801	0.250	0.250	0.250	4
SN024822	0.000	0.000	0.000	1
SN024823	0.000	0.000	0.000	3
SN024824	0.000	0.000	0.000	9
SN024825	0.000	0.000	0.000	3
SN024980	0.200	0.500	0.286	2
SN024981	0.500	0.167	0.250	6
SN024984	0.250	0.167	0.200	6
SN025079	1.000	0.333	0.500	3
SN025080	1.000	0.667	0.800	15
SN025081	0.357	0.556	0.435	9
SN025082	0.400	0.286	0.333	7
SN025083	0.727	0.533	0.615	15
SN025084	0.375	0.375	0.375	8
SN026227	0.600	0.273	0.375	11
SN026228	1.000	0.500	0.667	8
accuracy			0.200	700
macro avg	0.152	0.155	0.139	700
weighted avg	0.226	0.200	0.196	700
End of Table				

Table 21. Precision, Recall, Accuracy, and F1-Scores for Random Forest Classifier trained on EProbe Data.

Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN014855	0.500	0.800	0.615	5
SN014856	0.800	0.500	0.615	8
SN014857	0.444	1.000	0.615	4
SN016837	0.000	0.000	0.000	1
SN016838	0.200	0.125	0.154	8
SN016839	0.105	0.286	0.154	7
SN016841	0.000	0.000	0.000	9
SN016842	0.071	0.077	0.074	13

Continuation of Table 21				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN016843	0.000	0.000	0.000	1
SN016844	0.000	0.000	0.000	9
SN016845	0.211	0.308	0.250	13
SN016846	0.000	0.000	0.000	2
SN016847	0.400	0.200	0.267	10
SN016848	0.000	0.000	0.000	9
SN016849	0.154	0.286	0.200	7
SN016850	0.000	0.000	0.000	3
SN017952	0.500	0.100	0.167	10
SN017953	0.059	0.143	0.083	7
SN017955	0.429	0.500	0.462	6
SN017956	0.375	0.500	0.429	6
SN017959	0.833	0.556	0.667	9
SN018062	0.000	0.000	0.000	1
SN018064	0.000	0.000	0.000	11
SN018065	0.333	0.500	0.400	4
SN018066	0.000	0.000	0.000	1
SN018067	0.000	0.000	0.000	0
SN018068	0.500	0.286	0.364	7
SN018069	0.000	0.000	0.000	4
SN018070	0.000	0.000	0.000	3
SN018071	0.000	0.000	0.000	1
SN018072	0.333	0.500	0.400	2
SN018800	0.000	0.000	0.000	1
SN018802	0.500	0.091	0.154	11
SN018803	0.500	0.071	0.125	14
SN018804	0.000	0.000	0.000	3
SN018805	0.250	0.111	0.154	9
SN018806	0.833	0.714	0.769	7
SN018807	0.333	0.286	0.308	7
SN018808	0.143	0.250	0.182	4
SN018809	0.222	0.286	0.250	7
SN018810	0.000	0.000	0.000	6
SN018811	1.000	0.714	0.833	7

Continuation of Table 21				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN018812	0.333	0.143	0.200	7
SN019683	0.250	0.125	0.167	8
SN019684	0.333	0.111	0.167	9
SN019685	0.250	0.143	0.182	7
SN019688	0.000	0.000	0.000	17
SN020087	0.000	0.000	0.000	0
SN020089	0.000	0.000	0.000	1
SN020090	0.000	0.000	0.000	3
SN020091	0.000	0.000	0.000	2
SN020092	0.000	0.000	0.000	3
SN020093	0.000	0.000	0.000	1
SN020100	0.000	0.000	0.000	2
SN020101	0.000	0.000	0.000	1
SN020102	0.571	0.444	0.500	9
SN020104	0.200	0.333	0.250	3
SN020105	0.000	0.000	0.000	4
SN020107	0.200	0.500	0.286	4
SN020109	0.000	0.000	0.000	5
SN020885	0.000	0.000	0.000	0
SN020886	0.000	0.000	0.000	1
SN020887	0.000	0.000	0.000	2
SN020888	0.000	0.000	0.000	1
SN020889	0.000	0.000	0.000	2
SN020890	0.412	0.583	0.483	12
SN020891	0.000	0.000	0.000	9
SN020892	0.200	0.167	0.182	6
SN020893	0.286	0.400	0.333	5
SN022017	1.000	0.250	0.400	4
SN022018	0.750	0.333	0.462	9
SN022312	0.000	0.000	0.000	0
SN022313	0.000	0.000	0.000	2
SN022314	0.000	0.000	0.000	1
SN022315	0.000	0.000	0.000	3
SN022316	0.400	0.222	0.286	9

Continuation of Table 21				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN022457	0.000	0.000	0.000	0
SN022458	0.000	0.000	0.000	4
SN022459	0.375	0.375	0.375	8
SN022460	0.000	0.000	0.000	2
SN022464	0.000	0.000	0.000	2
SN022817	0.222	0.500	0.308	4
SN022818	0.235	0.800	0.364	5
SN022819	0.000	0.000	0.000	1
SN022823	0.000	0.000	0.000	3
SN023170	0.000	0.000	0.000	3
SN023172	0.000	0.000	0.000	1
SN023175	0.000	0.000	0.000	6
SN023176	0.800	0.400	0.533	10
SN023177	0.250	0.286	0.267	7
SN023178	0.000	0.000	0.000	2
SN023179	0.167	0.333	0.222	3
SN023180	0.062	0.167	0.091	6
SN023181	0.444	0.667	0.533	6
SN023182	0.000	0.000	0.000	4
SN023183	0.400	0.667	0.500	6
SN023185	0.333	0.200	0.250	5
SN023186	0.000	0.000	0.000	4
SN023523	0.333	0.333	0.333	6
SN023524	0.545	0.545	0.545	11
SN023527	0.000	0.000	0.000	5
SN023531	0.143	0.125	0.133	8
SN023535	0.000	0.000	0.000	5
SN023538	0.000	0.000	0.000	2
SN024399	0.000	0.000	0.000	1
SN024400	0.000	0.000	0.000	4
SN024401	0.000	0.000	0.000	5
SN024402	0.333	0.143	0.200	7
SN024794	0.000	0.000	0.000	4
SN024795	0.000	0.000	0.000	4

Continuation of Table 21				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	8
SN024798	0.000	0.000	0.000	2
SN024799	0.000	0.000	0.000	9
SN024800	0.286	0.286	0.286	7
SN024801	0.200	0.500	0.286	4
SN024822	0.000	0.000	0.000	1
SN024823	0.000	0.000	0.000	3
SN024824	0.125	0.222	0.160	9
SN024825	0.000	0.000	0.000	3
SN024980	0.500	1.000	0.667	2
SN024981	0.667	0.333	0.444	6
SN024984	0.600	0.500	0.545	6
SN025079	1.000	0.333	0.500	3
SN025080	0.650	0.867	0.743	15
SN025081	0.500	0.556	0.526	9
SN025082	0.500	0.429	0.462	7
SN025083	0.750	0.600	0.667	15
SN025084	0.400	0.500	0.444	8
SN026227	0.444	0.364	0.400	11
SN026228	1.000	0.500	0.667	8
accuracy			0.256	700
macro avg	0.210	0.194	0.183	700
weighted avg	0.299	0.256	0.253	700
End of Table				

Table 22. Precision, Recall, Accuracy, and F1-Scores for Extra Forest Classifier trained on EProbe Data.

Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN014855	0.500	0.800	0.615	5
SN014856	0.500	0.500	0.500	8
SN014857	0.444	1.000	0.615	4

Continuation of Table 22				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN016837	0.000	0.000	0.000	1
SN016838	0.286	0.250	0.267	8
SN016839	0.125	0.143	0.133	7
SN016841	0.143	0.111	0.125	9
SN016842	0.222	0.308	0.258	13
SN016843	0.000	0.000	0.000	1
SN016844	0.250	0.111	0.154	9
SN016845	0.250	0.231	0.240	13
SN016846	0.000	0.000	0.000	2
SN016847	0.250	0.200	0.222	10
SN016848	0.000	0.000	0.000	9
SN016849	0.154	0.286	0.200	7
SN016850	0.000	0.000	0.000	3
SN017952	0.556	0.500	0.526	10
SN017953	0.154	0.286	0.200	7
SN017955	0.444	0.667	0.533	6
SN017956	0.333	0.500	0.400	6
SN017959	0.625	0.556	0.588	9
SN018062	0.000	0.000	0.000	1
SN018064	0.667	0.364	0.471	11
SN018065	0.500	0.500	0.500	4
SN018066	0.000	0.000	0.000	1
SN018067	0.000	0.000	0.000	0
SN018068	0.333	0.286	0.308	7
SN018069	0.000	0.000	0.000	4
SN018070	0.000	0.000	0.000	3
SN018071	0.000	0.000	0.000	1
SN018072	0.167	0.500	0.250	2
SN018800	0.000	0.000	0.000	1
SN018802	0.500	0.182	0.267	11
SN018803	0.333	0.071	0.118	14
SN018804	0.000	0.000	0.000	3
SN018805	0.500	0.333	0.400	9
SN018806	0.667	0.857	0.750	7

Continuation of Table 22				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN018807	1.000	0.429	0.600	7
SN018808	0.091	0.250	0.133	4
SN018809	0.222	0.286	0.250	7
SN018810	0.000	0.000	0.000	6
SN018811	0.833	0.714	0.769	7
SN018812	0.250	0.143	0.182	7
SN019683	0.200	0.125	0.154	8
SN019684	0.167	0.111	0.133	9
SN019685	0.000	0.000	0.000	7
SN019688	0.500	0.235	0.320	17
SN020087	0.000	0.000	0.000	0
SN020088	0.000	0.000	0.000	0
SN020089	0.000	0.000	0.000	1
SN020090	0.000	0.000	0.000	3
SN020091	0.000	0.000	0.000	2
SN020092	0.000	0.000	0.000	3
SN020093	0.000	0.000	0.000	1
SN020100	0.000	0.000	0.000	2
SN020101	0.000	0.000	0.000	1
SN020102	0.500	0.222	0.308	9
SN020104	0.500	0.333	0.400	3
SN020105	0.000	0.000	0.000	4
SN020107	0.286	0.500	0.364	4
SN020109	0.000	0.000	0.000	5
SN020886	0.000	0.000	0.000	1
SN020887	0.000	0.000	0.000	2
SN020888	0.000	0.000	0.000	1
SN020889	0.000	0.000	0.000	2
SN020890	0.429	0.500	0.462	12
SN020891	0.000	0.000	0.000	9
SN020892	0.167	0.167	0.167	6
SN020893	0.400	0.400	0.400	5
SN022017	0.000	0.000	0.000	4
SN022018	0.500	0.333	0.400	9

Continuation of Table 22				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN022312	0.000	0.000	0.000	0
SN022313	0.000	0.000	0.000	2
SN022314	0.000	0.000	0.000	1
SN022315	0.000	0.000	0.000	3
SN022316	0.143	0.111	0.125	9
SN022458	0.000	0.000	0.000	4
SN022459	0.444	0.500	0.471	8
SN022460	0.000	0.000	0.000	2
SN022464	0.000	0.000	0.000	2
SN022817	0.333	0.500	0.400	4
SN022818	0.267	0.800	0.400	5
SN022819	0.000	0.000	0.000	1
SN022823	0.000	0.000	0.000	3
SN023170	0.000	0.000	0.000	3
SN023172	0.000	0.000	0.000	1
SN023174	0.000	0.000	0.000	0
SN023175	0.250	0.167	0.200	6
SN023176	0.300	0.300	0.300	10
SN023177	0.286	0.286	0.286	7
SN023178	0.000	0.000	0.000	2
SN023179	0.143	0.333	0.200	3
SN023180	0.100	0.167	0.125	6
SN023181	0.375	0.500	0.429	6
SN023182	0.000	0.000	0.000	4
SN023183	0.462	1.000	0.632	6
SN023185	0.000	0.000	0.000	5
SN023186	0.000	0.000	0.000	4
SN023522	0.000	0.000	0.000	0
SN023523	0.167	0.167	0.167	6
SN023524	0.444	0.364	0.400	11
SN023527	0.000	0.000	0.000	5
SN023528	0.000	0.000	0.000	0
SN023531	0.286	0.250	0.267	8
SN023535	0.000	0.000	0.000	5

Continuation of Table 22				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN023538	0.000	0.000	0.000	2
SN024399	0.000	0.000	0.000	1
SN024400	0.000	0.000	0.000	4
SN024401	0.000	0.000	0.000	5
SN024402	0.333	0.143	0.200	7
SN024794	0.000	0.000	0.000	4
SN024795	0.125	0.250	0.167	4
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	8
SN024798	0.250	0.500	0.333	2
SN024799	0.000	0.000	0.000	9
SN024800	0.429	0.429	0.429	7
SN024801	0.083	0.250	0.125	4
SN024822	0.000	0.000	0.000	1
SN024823	0.000	0.000	0.000	3
SN024824	0.200	0.222	0.211	9
SN024825	0.000	0.000	0.000	3
SN024980	0.125	0.500	0.200	2
SN024981	0.667	0.333	0.444	6
SN024984	0.800	0.667	0.727	6
SN025079	0.500	0.333	0.400	3
SN025080	0.619	0.867	0.722	15
SN025081	0.600	0.667	0.632	9
SN025082	0.500	0.429	0.462	7
SN025083	0.833	0.667	0.741	15
SN025084	0.556	0.625	0.588	8
SN026227	0.750	0.273	0.400	11
SN026228	1.000	0.500	0.667	8
accuracy			0.284	700
macro avg	0.205	0.206	0.192	700
weighted avg	0.314	0.284	0.281	700
End of Table				

Table 23. Precision, Recall, Accuracy, and F1-Scores for Voting Classifier with all models trained on EProbe Data.

Sample ID	Precision	Recall	F1-Score	Support
SN014855	0.833	1.000	0.909	5
SN014856	0.556	0.625	0.588	8
SN014857	0.400	1.000	0.571	4
SN016837	0.000	0.000	0.000	1
SN016838	0.000	0.000	0.000	8
SN016839	0.400	0.286	0.333	7
SN016840	0.000	0.000	0.000	0
SN016841	0.000	0.000	0.000	9
SN016842	0.200	0.154	0.174	13
SN016843	0.000	0.000	0.000	1
SN016844	0.000	0.000	0.000	9
SN016845	0.273	0.231	0.250	13
SN016846	0.167	0.500	0.250	2
SN016847	0.400	0.200	0.267	10
SN016848	0.143	0.111	0.125	9
SN016849	0.273	0.429	0.333	7
SN016850	0.125	0.333	0.182	3
SN017952	0.000	0.000	0.000	10
SN017953	0.000	0.000	0.000	7
SN017955	0.333	0.500	0.400	6
SN017956	0.273	0.500	0.353	6
SN017959	1.000	0.556	0.714	9
SN018062	0.000	0.000	0.000	1
SN018064	0.000	0.000	0.000	11
SN018065	0.500	0.500	0.500	4
SN018066	0.000	0.000	0.000	1
SN018068	0.333	0.143	0.200	7
SN018069	0.000	0.000	0.000	4
SN018070	0.000	0.000	0.000	3
SN018071	0.000	0.000	0.000	1
SN018072	0.333	0.500	0.400	2
SN018800	0.000	0.000	0.000	1
SN018802	0.333	0.091	0.143	11

Continuation of Table 23				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN018803	0.000	0.000	0.000	14
SN018804	0.000	0.000	0.000	3
SN018805	0.000	0.000	0.000	9
SN018806	0.714	0.714	0.714	7
SN018807	0.188	0.429	0.261	7
SN018808	0.000	0.000	0.000	4
SN018809	0.286	0.571	0.381	7
SN018810	0.143	0.167	0.154	6
SN018811	1.000	0.714	0.833	7
SN018812	0.500	0.143	0.222	7
SN019683	0.333	0.125	0.182	8
SN019684	0.231	0.333	0.273	9
SN019685	0.000	0.000	0.000	7
SN019688	0.200	0.059	0.091	17
SN020087	0.000	0.000	0.000	0
SN020088	0.000	0.000	0.000	0
SN020089	0.000	0.000	0.000	1
SN020090	0.000	0.000	0.000	3
SN020091	0.000	0.000	0.000	2
SN020092	0.000	0.000	0.000	3
SN020093	0.000	0.000	0.000	1
SN020100	0.000	0.000	0.000	2
SN020101	0.000	0.000	0.000	1
SN020102	0.333	0.222	0.267	9
SN020104	0.333	0.333	0.333	3
SN020105	0.000	0.000	0.000	4
SN020107	0.111	0.250	0.154	4
SN020109	0.000	0.000	0.000	5
SN020885	0.000	0.000	0.000	0
SN020886	0.000	0.000	0.000	1
SN020887	0.000	0.000	0.000	2
SN020888	0.000	0.000	0.000	1
SN020889	0.000	0.000	0.000	2
SN020890	0.400	0.500	0.444	12

Continuation of Table 23				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN020891	0.000	0.000	0.000	9
SN020892	0.111	0.167	0.133	6
SN020893	0.154	0.400	0.222	5
SN022017	0.500	0.250	0.333	4
SN022018	1.000	0.222	0.364	9
SN022312	0.000	0.000	0.000	0
SN022313	0.111	0.500	0.182	2
SN022314	0.000	0.000	0.000	1
SN022315	0.000	0.000	0.000	3
SN022316	0.200	0.222	0.211	9
SN022457	0.000	0.000	0.000	0
SN022458	0.000	0.000	0.000	4
SN022459	0.667	0.500	0.571	8
SN022460	0.000	0.000	0.000	2
SN022464	0.000	0.000	0.000	2
SN022817	0.286	0.500	0.364	4
SN022818	0.286	0.800	0.421	5
SN022819	0.000	0.000	0.000	1
SN022823	0.000	0.000	0.000	3
SN023170	0.000	0.000	0.000	3
SN023172	0.000	0.000	0.000	1
SN023175	0.000	0.000	0.000	6
SN023176	0.250	0.300	0.273	10
SN023177	0.333	0.143	0.200	7
SN023178	0.000	0.000	0.000	2
SN023179	0.333	0.667	0.444	3
SN023180	0.167	0.167	0.167	6
SN023181	0.429	0.500	0.462	6
SN023182	0.000	0.000	0.000	4
SN023183	0.600	1.000	0.750	6
SN023185	0.000	0.000	0.000	5
SN023186	0.000	0.000	0.000	4
SN023523	0.200	0.167	0.182	6
SN023524	0.200	0.273	0.231	11

Continuation of Table 23				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN023527	0.000	0.000	0.000	5
SN023528	0.000	0.000	0.000	0
SN023531	0.250	0.125	0.167	8
SN023535	0.000	0.000	0.000	5
SN023538	0.182	1.000	0.308	2
SN024399	0.000	0.000	0.000	1
SN024400	0.000	0.000	0.000	4
SN024401	0.000	0.000	0.000	5
SN024402	0.000	0.000	0.000	7
SN024794	0.000	0.000	0.000	4
SN024795	0.000	0.000	0.000	4
SN024796	0.000	0.000	0.000	1
SN024797	0.250	0.125	0.167	8
SN024798	0.000	0.000	0.000	2
SN024799	0.000	0.000	0.000	9
SN024800	0.133	0.286	0.182	7
SN024801	0.250	0.250	0.250	4
SN024822	0.000	0.000	0.000	1
SN024823	0.000	0.000	0.000	3
SN024824	0.000	0.000	0.000	9
SN024825	0.000	0.000	0.000	3
SN024980	0.333	0.500	0.400	2
SN024981	0.429	0.500	0.462	6
SN024984	0.500	0.333	0.400	6
SN025079	0.500	0.333	0.400	3
SN025080	0.750	0.800	0.774	15
SN025081	0.417	0.556	0.476	9
SN025082	0.250	0.143	0.182	7
SN025083	0.500	0.533	0.516	15
SN025084	0.375	0.375	0.375	8
SN026227	0.625	0.455	0.526	11
SN026228	1.000	0.500	0.667	8
accuracy			0.243	700
macro avg	0.182	0.194	0.171	700

Continuation of Table 23				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
weighted avg	0.258	0.243	0.231	700
End of Table				

Table 24. Precision, Recall, Accuracy, and F1-Scores for Voting Classifier with tree models only trained on EProbe Data.

Sample ID	Precision	Recall	F1-Score	Support
SN014855	0.500	0.800	0.615	5
SN014856	0.571	0.500	0.533	8
SN014857	0.444	1.000	0.615	4
SN016837	0.000	0.000	0.000	1
SN016838	0.333	0.125	0.182	8
SN016839	0.200	0.143	0.167	7
SN016841	0.000	0.000	0.000	9
SN016842	0.150	0.231	0.182	13
SN016843	0.000	0.000	0.000	1
SN016844	0.250	0.111	0.154	9
SN016845	0.273	0.231	0.250	13
SN016846	0.000	0.000	0.000	2
SN016847	0.333	0.200	0.250	10
SN016848	0.000	0.000	0.000	9
SN016849	0.154	0.286	0.200	7
SN016850	0.125	0.333	0.182	3
SN017952	0.667	0.400	0.500	10
SN017953	0.000	0.000	0.000	7
SN017955	0.444	0.667	0.533	6
SN017956	0.250	0.333	0.286	6
SN017959	0.714	0.556	0.625	9
SN018062	0.000	0.000	0.000	1
SN018064	0.400	0.364	0.381	11
SN018065	0.500	0.500	0.500	4
SN018066	0.000	0.000	0.000	1
SN018067	0.000	0.000	0.000	0

Continuation of Table 24				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN018068	0.667	0.286	0.400	7
SN018069	0.000	0.000	0.000	4
SN018070	0.000	0.000	0.000	3
SN018071	0.000	0.000	0.000	1
SN018072	0.200	0.500	0.286	2
SN018800	0.000	0.000	0.000	1
SN018802	0.600	0.273	0.375	11
SN018803	0.000	0.000	0.000	14
SN018804	0.000	0.000	0.000	3
SN018805	0.500	0.556	0.526	9
SN018806	0.833	0.714	0.769	7
SN018807	0.667	0.571	0.615	7
SN018808	0.125	0.250	0.167	4
SN018809	0.222	0.286	0.250	7
SN018810	0.000	0.000	0.000	6
SN018811	0.714	0.714	0.714	7
SN018812	0.333	0.286	0.308	7
SN019683	0.250	0.125	0.167	8
SN019684	0.100	0.111	0.105	9
SN019685	0.333	0.143	0.200	7
SN019688	0.333	0.059	0.100	17
SN020087	0.000	0.000	0.000	0
SN020088	0.000	0.000	0.000	0
SN020089	0.000	0.000	0.000	1
SN020090	0.000	0.000	0.000	3
SN020091	0.000	0.000	0.000	2
SN020092	0.000	0.000	0.000	3
SN020093	0.000	0.000	0.000	1
SN020100	0.000	0.000	0.000	2
SN020101	0.000	0.000	0.000	1
SN020102	0.429	0.333	0.375	9
SN020104	0.250	0.333	0.286	3
SN020105	0.000	0.000	0.000	4
SN020107	0.200	0.500	0.286	4

Continuation of Table 24				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN020109	0.000	0.000	0.000	5
SN020885	0.000	0.000	0.000	0
SN020886	0.000	0.000	0.000	1
SN020887	0.000	0.000	0.000	2
SN020888	0.000	0.000	0.000	1
SN020889	0.000	0.000	0.000	2
SN020890	0.500	0.583	0.538	12
SN020891	0.000	0.000	0.000	9
SN020892	0.250	0.167	0.200	6
SN020893	0.333	0.400	0.364	5
SN022017	0.500	0.250	0.333	4
SN022018	0.429	0.333	0.375	9
SN022312	0.000	0.000	0.000	0
SN022313	0.000	0.000	0.000	2
SN022314	0.000	0.000	0.000	1
SN022315	0.000	0.000	0.000	3
SN022316	0.333	0.111	0.167	9
SN022457	0.000	0.000	0.000	0
SN022458	0.000	0.000	0.000	4
SN022459	0.429	0.375	0.400	8
SN022460	0.000	0.000	0.000	2
SN022464	0.000	0.000	0.000	2
SN022817	0.333	0.500	0.400	4
SN022818	0.267	0.800	0.400	5
SN022819	0.000	0.000	0.000	1
SN022823	0.000	0.000	0.000	3
SN023170	0.000	0.000	0.000	3
SN023172	0.000	0.000	0.000	1
SN023175	0.143	0.167	0.154	6
SN023176	0.500	0.100	0.167	10
SN023177	0.333	0.286	0.308	7
SN023178	0.000	0.000	0.000	2
SN023179	0.250	0.667	0.364	3
SN023180	0.067	0.167	0.095	6

Continuation of Table 24				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN023181	0.400	0.667	0.500	6
SN023182	0.000	0.000	0.000	4
SN023183	0.462	1.000	0.632	6
SN023185	0.250	0.200	0.222	5
SN023186	0.000	0.000	0.000	4
SN023523	0.143	0.167	0.154	6
SN023524	0.333	0.273	0.300	11
SN023527	0.000	0.000	0.000	5
SN023531	0.000	0.000	0.000	8
SN023535	0.000	0.000	0.000	5
SN023538	0.000	0.000	0.000	2
SN024399	0.000	0.000	0.000	1
SN024400	0.000	0.000	0.000	4
SN024401	0.000	0.000	0.000	5
SN024402	0.333	0.143	0.200	7
SN024794	0.000	0.000	0.000	4
SN024795	0.100	0.250	0.143	4
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	8
SN024798	0.250	0.500	0.333	2
SN024799	0.000	0.000	0.000	9
SN024800	0.333	0.429	0.375	7
SN024801	0.167	0.500	0.250	4
SN024822	0.000	0.000	0.000	1
SN024823	0.200	0.333	0.250	3
SN024824	0.200	0.222	0.211	9
SN024825	0.000	0.000	0.000	3
SN024980	0.286	1.000	0.444	2
SN024981	0.750	0.500	0.600	6
SN024984	0.600	0.500	0.545	6
SN025079	0.500	0.333	0.400	3
SN025080	0.650	0.867	0.743	15
SN025081	0.538	0.778	0.636	9
SN025082	0.500	0.429	0.462	7

Continuation of Table 24				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN025083	0.818	0.600	0.692	15
SN025084	0.500	0.625	0.556	8
SN026227	0.800	0.364	0.500	11
SN026228	1.000	0.500	0.667	8
accuracy			0.283	700
macro avg	0.211	0.219	0.198	700
weighted avg	0.308	0.283	0.275	700
End of Table				

Table 25. Precision, Recall, Accuracy, and F1-Scores for Logistic Regression trained on TESCAN Data.

Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN023729	0.122	0.736	0.209	258
SN023730	0.864	0.442	0.585	6009
SN023731	0.375	0.628	0.470	1789
SN025331	0.429	0.714	0.536	1004
SN025332	0.708	0.801	0.752	7627
SN025333	0.191	0.112	0.142	1006
SN025334	0.056	0.587	0.103	167
SN025335	0.157	0.041	0.065	533
SN025336	0.592	0.282	0.382	5439
SN025337	0.804	0.893	0.846	7554
accuracy			0.615	31386
macro avg	0.430	0.524	0.409	31386
weighted avg	0.679	0.615	0.616	31386
End of Table				

Table 26. Precision, Recall, Accuracy, and F1-Scores for Linear Discriminate Analysis trained on TESCAN Data.

Sample ID	Precision	Recall	F1-Score	Support
SN023729	0.161	0.655	0.258	258
SN023730	0.772	0.578	0.661	6009
SN023731	0.387	0.234	0.292	1789
SN025331	0.384	0.696	0.495	1004
SN025332	0.654	0.840	0.735	7627
SN025333	0.102	0.063	0.077	1006
SN025334	0.075	0.299	0.120	167
SN025335	0.148	0.023	0.039	533
SN025336	0.513	0.170	0.256	5439
SN025337	0.695	0.917	0.790	7554
accuracy			0.610	31386
macro avg	0.389	0.448	0.372	31386
weighted avg	0.605	0.610	0.578	31386
End of Table				

Table 27. Precision, Recall, Accuracy, and F1-Scores for Decision Tree Classifier trained on TESCAN Data.

Sample ID	Precision	Recall	F1-Score	Support
SN023729	0.408	0.345	0.374	258
SN023730	0.815	0.820	0.818	6009
SN023731	0.517	0.516	0.517	1789
SN025331	0.536	0.541	0.538	1004
SN025332	0.757	0.746	0.752	7627
SN025333	0.301	0.343	0.321	1006
SN025334	0.108	0.096	0.102	167
SN025335	0.169	0.161	0.165	533
SN025336	0.556	0.557	0.557	5439
SN025337	0.855	0.855	0.855	7554
accuracy			0.704	31386

Continuation of Table 27				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
macro avg	0.502	0.498	0.500	31386
weighted avg	0.705	0.704	0.705	31386
End of Table				

Table 28. Precision, Recall, Accuracy, and F1-Scores for Random Forest Classifier trained on TESCAN Data.

Sample ID	Precision	Recall	F1-Score	Support
SN023729	0.568	0.341	0.426	258
SN023730	0.824	0.936	0.876	6009
SN023731	0.761	0.499	0.603	1789
SN025331	0.582	0.694	0.633	1004
SN025332	0.795	0.844	0.819	7627
SN025333	0.401	0.376	0.388	1006
SN025334	0.067	0.018	0.028	167
SN025335	0.299	0.184	0.228	533
SN025336	0.691	0.585	0.634	5439
SN025337	0.874	0.927	0.900	7554
accuracy			0.778	31386
macro avg	0.586	0.540	0.553	31386
weighted avg	0.766	0.778	0.768	31386
End of Table				

Table 29. Precision, Recall, Accuracy, and F1-Scores for Extra Forest Classifier trained on TESCAN Data.

Sample ID	Precision	Recall	F1-Score	Support
SN023729	0.590	0.395	0.473	258
SN023730	0.823	0.936	0.876	6009
SN023731	0.756	0.477	0.585	1789

Continuation of Table 29				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN025331	0.595	0.676	0.633	1004
SN025332	0.790	0.836	0.812	7627
SN025333	0.412	0.370	0.390	1006
SN025334	0.200	0.120	0.150	167
SN025335	0.298	0.210	0.246	533
SN025336	0.685	0.586	0.632	5439
SN025337	0.874	0.927	0.900	7554
accuracy			0.775	31386
macro avg	0.602	0.553	0.570	31386
weighted avg	0.765	0.775	0.766	31386
End of Table				

Table 30. Precision, Recall, Accuracy, and F1-Scores for Voting Classifier with all models trained on TESCAN Data.

Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN023729	0.439	0.516	0.474	258
SN023730	0.833	0.889	0.860	6009
SN023731	0.679	0.505	0.579	1789
SN025331	0.512	0.741	0.605	1004
SN025332	0.779	0.844	0.810	7627
SN025333	0.403	0.290	0.337	1006
SN025334	0.148	0.180	0.162	167
SN025335	0.257	0.144	0.185	533
SN025336	0.694	0.528	0.600	5439
SN025337	0.856	0.936	0.894	7554
accuracy			0.761	31386
macro avg	0.560	0.557	0.551	31386
weighted avg	0.752	0.761	0.752	31386
End of Table				

Table 31. Precision, Recall, Accuracy, and F1-Scores for Voting Classifier with tree models only trained on TESCAN Data.

Sample ID	Precision	Recall	F1-Score	Support
SN023729	0.599	0.388	0.471	258
SN023730	0.826	0.940	0.879	6009
SN023731	0.769	0.490	0.598	1789
SN025331	0.603	0.697	0.647	1004
SN025332	0.796	0.842	0.818	7627
SN025333	0.409	0.376	0.392	1006
SN025334	0.224	0.114	0.151	167
SN025335	0.315	0.203	0.247	533
SN025336	0.694	0.592	0.639	5439
SN025337	0.877	0.930	0.902	7554
accuracy			0.780	31386
macro avg	0.611	0.557	0.574	31386
weighted avg	0.770	0.780	0.771	31386
End of Table				

Table 32. Precision, Recall, Accuracy, and F1-Scores for Logistic Regression trained on both datasets.

Sample ID	Precision	Recall	F1-Score	Support
SN014855	0.062	0.111	0.080	9
SN014856	0.000	0.000	0.000	4
SN014857	0.024	1.000	0.047	5
SN016837	0.000	0.000	0.000	1
SN016838	0.012	0.125	0.021	8
SN016839	0.035	0.375	0.064	8
SN016840	0.000	0.000	0.000	1
SN016841	0.000	0.000	0.000	9
SN016842	0.000	0.000	0.000	6
SN016843	0.000	0.000	0.000	3

Continuation of Table 32				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN016844	0.000	0.000	0.000	9
SN016845	0.000	0.000	0.000	7
SN016846	0.000	0.000	0.000	4
SN016847	0.026	0.111	0.042	9
SN016848	0.018	0.200	0.034	10
SN016849	0.000	0.000	0.000	6
SN016850	0.034	0.250	0.061	4
SN017952	0.833	0.455	0.588	11
SN017953	0.000	0.000	0.000	4
SN017955	0.038	0.167	0.062	6
SN017956	0.167	0.286	0.211	7
SN017959	0.030	1.000	0.058	5
SN018064	0.143	0.143	0.143	7
SN018065	0.300	1.000	0.462	3
SN018066	0.000	0.000	0.000	1
SN018067	0.000	0.000	0.000	0
SN018068	0.250	0.500	0.333	6
SN018069	0.000	0.000	0.000	3
SN018070	0.000	0.000	0.000	2
SN018071	0.000	0.000	0.000	2
SN018072	0.048	0.500	0.087	2
SN018800	0.000	0.000	0.000	1
SN018802	0.143	0.231	0.176	13
SN018803	0.098	0.800	0.174	5
SN018804	0.002	0.200	0.005	5
SN018805	0.133	0.400	0.200	5
SN018806	0.188	0.273	0.222	11
SN018807	0.091	0.429	0.150	7
SN018808	0.000	0.000	0.000	5
SN018809	0.062	0.143	0.087	7
SN018810	0.333	0.167	0.222	6
SN018811	0.086	0.333	0.136	9
SN018812	0.000	0.000	0.000	5
SN019683	0.016	0.111	0.027	9

Continuation of Table 32				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN019684	0.004	0.200	0.007	10
SN019685	0.167	0.167	0.167	6
SN019688	0.050	0.429	0.089	14
SN020087	0.000	0.000	0.000	2
SN020088	0.000	0.000	0.000	0
SN020089	0.000	0.000	0.000	2
SN020090	0.000	0.000	0.000	2
SN020091	0.000	0.000	0.000	1
SN020092	0.000	0.000	0.000	2
SN020093	0.000	0.000	0.000	3
SN020100	0.000	0.000	0.000	0
SN020101	0.000	0.000	0.000	2
SN020102	0.000	0.000	0.000	9
SN020104	0.100	0.200	0.133	5
SN020105	0.000	0.000	0.000	5
SN020107	0.019	0.200	0.035	5
SN020109	0.000	0.000	0.000	4
SN020885	0.000	0.000	0.000	1
SN020886	0.000	0.000	0.000	0
SN020887	0.000	0.000	0.000	1
SN020888	0.000	0.000	0.000	1
SN020889	0.125	0.125	0.125	8
SN020890	0.008	0.167	0.015	12
SN020891	0.029	0.333	0.053	3
SN020892	0.000	0.000	0.000	6
SN020893	0.056	0.400	0.098	5
SN022017	0.000	0.000	0.000	5
SN022018	0.100	0.375	0.158	8
SN022312	0.000	0.000	0.000	0
SN022313	0.250	0.400	0.308	5
SN022314	0.000	0.000	0.000	2
SN022315	0.000	0.000	0.000	2
SN022316	0.167	0.125	0.143	8
SN022456	0.000	0.000	0.000	1

Continuation of Table 32				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN022457	0.000	0.000	0.000	1
SN022458	0.000	0.000	0.000	5
SN022459	0.002	0.125	0.005	8
SN022460	0.143	0.500	0.222	2
SN022464	0.000	0.000	0.000	3
SN022817	0.000	0.000	0.000	10
SN022818	0.069	0.667	0.125	3
SN022819	0.000	0.000	0.000	0
SN022823	0.009	0.333	0.018	3
SN023170	0.000	0.000	0.000	5
SN023172	0.000	0.000	0.000	0
SN023174	0.000	0.000	0.000	0
SN023175	0.000	0.000	0.000	4
SN023176	0.000	0.000	0.000	5
SN023177	0.200	0.167	0.182	6
SN023178	0.000	0.000	0.000	3
SN023179	0.143	0.200	0.167	5
SN023180	0.013	0.143	0.023	7
SN023181	0.600	0.750	0.667	4
SN023182	0.000	0.000	0.000	1
SN023183	0.013	1.000	0.025	1
SN023185	0.000	0.000	0.000	8
SN023186	0.000	0.000	0.000	2
SN023522	0.000	0.000	0.000	1
SN023523	0.286	0.167	0.211	12
SN023524	0.154	0.222	0.182	9
SN023527	0.000	0.000	0.000	3
SN023528	0.000	0.000	0.000	1
SN023531	0.000	0.000	0.000	10
SN023535	0.000	0.000	0.000	4
SN023538	0.000	0.000	0.000	6
SN023729	0.120	0.632	0.202	223
SN023730	0.810	0.181	0.296	6082
SN023731	0.270	0.136	0.181	1737

Continuation of Table 32				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN024400	0.200	0.167	0.182	6
SN024401	0.000	0.000	0.000	9
SN024402	0.000	0.000	0.000	8
SN024794	0.143	0.250	0.182	4
SN024795	0.000	0.000	0.000	7
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	9
SN024798	0.000	0.000	0.000	3
SN024799	0.125	0.091	0.105	11
SN024800	0.059	0.111	0.077	9
SN024801	0.000	0.000	0.000	6
SN024822	0.000	0.000	0.000	0
SN024823	0.000	0.000	0.000	2
SN024824	0.000	0.000	0.000	5
SN024825	0.000	0.000	0.000	6
SN024980	0.000	0.000	0.000	3
SN024981	0.067	0.500	0.118	2
SN024984	0.028	1.000	0.054	2
SN025079	0.000	0.000	0.000	3
SN025080	0.071	0.462	0.124	13
SN025081	0.024	0.333	0.045	6
SN025082	0.039	0.286	0.069	7
SN025083	0.018	0.429	0.034	14
SN025084	0.071	0.250	0.111	12
SN025331	0.433	0.695	0.533	1012
SN025332	0.719	0.752	0.736	7603
SN025333	0.139	0.196	0.163	1004
SN025334	0.064	0.512	0.115	170
SN025335	0.056	0.034	0.042	530
SN025336	0.586	0.323	0.416	5465
SN025337	0.819	0.777	0.797	7572
SN026227	0.875	0.538	0.667	13
SN026228	0.333	1.000	0.500	4
accuracy			0.498	32085

Continuation of Table 32				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
macro avg	0.082	0.182	0.087	32085
weighted avg	0.653	0.498	0.526	32085
End of Table				

Table 33. Precision, Recall, Accuracy, and F1-Scores for Linear Discriminate Analysis trained on both datasets.

Sample ID	Precision	Recall	F1-Score	Support
SN014855	0.000	0.000	0.000	9
SN014856	0.091	0.250	0.133	4
SN014857	0.000	0.000	0.000	5
SN016837	0.000	0.000	0.000	1
SN016838	0.000	0.000	0.000	8
SN016839	0.000	0.000	0.000	8
SN016840	0.000	0.000	0.000	1
SN016841	0.000	0.000	0.000	9
SN016842	0.000	0.000	0.000	6
SN016843	0.000	0.000	0.000	3
SN016844	0.000	0.000	0.000	9
SN016845	0.000	0.000	0.000	7
SN016846	0.000	0.000	0.000	4
SN016847	0.000	0.000	0.000	9
SN016848	0.000	0.000	0.000	10
SN016849	0.000	0.000	0.000	6
SN016850	0.000	0.000	0.000	4
SN017952	0.000	0.000	0.000	11
SN017953	0.000	0.000	0.000	4
SN017955	0.000	0.000	0.000	6
SN017956	0.143	0.286	0.190	7
SN017959	0.000	0.000	0.000	5
SN018062	0.000	0.000	0.000	0
SN018064	0.000	0.000	0.000	7
SN018065	0.000	0.000	0.000	3

Continuation of Table 33				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN018066	0.000	0.000	0.000	1
SN018068	0.000	0.000	0.000	6
SN018069	0.000	0.000	0.000	3
SN018070	0.000	0.000	0.000	2
SN018071	0.000	0.000	0.000	2
SN018072	0.000	0.000	0.000	2
SN018800	0.000	0.000	0.000	1
SN018802	0.000	0.000	0.000	13
SN018803	0.000	0.000	0.000	5
SN018804	0.000	0.000	0.000	5
SN018805	0.000	0.000	0.000	5
SN018806	0.143	0.182	0.160	11
SN018807	0.000	0.000	0.000	7
SN018808	0.000	0.000	0.000	5
SN018809	0.000	0.000	0.000	7
SN018810	0.333	0.167	0.222	6
SN018811	0.167	0.222	0.190	9
SN018812	0.000	0.000	0.000	5
SN019683	0.000	0.000	0.000	9
SN019684	0.000	0.000	0.000	10
SN019685	0.000	0.000	0.000	6
SN019688	0.000	0.000	0.000	14
SN020087	0.011	0.500	0.022	2
SN020088	0.000	0.000	0.000	0
SN020089	0.000	0.000	0.000	2
SN020090	0.000	0.000	0.000	2
SN020091	0.000	0.000	0.000	1
SN020092	0.000	0.000	0.000	2
SN020093	0.000	0.000	0.000	3
SN020101	0.000	0.000	0.000	2
SN020102	0.200	0.111	0.143	9
SN020104	0.000	0.000	0.000	5
SN020105	0.000	0.000	0.000	5
SN020107	0.000	0.000	0.000	5

Continuation of Table 33				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN020109	0.000	0.000	0.000	4
SN020885	0.000	0.000	0.000	1
SN020887	0.000	0.000	0.000	1
SN020888	0.000	0.000	0.000	1
SN020889	0.000	0.000	0.000	8
SN020890	0.000	0.000	0.000	12
SN020891	0.000	0.000	0.000	3
SN020892	0.000	0.000	0.000	6
SN020893	0.000	0.000	0.000	5
SN022017	0.000	0.000	0.000	5
SN022018	0.000	0.000	0.000	8
SN022312	0.000	0.000	0.000	0
SN022313	0.012	0.200	0.022	5
SN022314	0.000	0.000	0.000	2
SN022315	0.000	0.000	0.000	2
SN022316	0.000	0.000	0.000	8
SN022456	0.000	0.000	0.000	1
SN022457	0.000	0.000	0.000	1
SN022458	0.000	0.000	0.000	5
SN022459	0.000	0.000	0.000	8
SN022460	0.000	0.000	0.000	2
SN022464	0.050	0.333	0.087	3
SN022817	0.000	0.000	0.000	10
SN022818	0.000	0.000	0.000	3
SN022819	0.000	0.000	0.000	0
SN022823	0.100	0.333	0.154	3
SN023170	0.000	0.000	0.000	5
SN023172	0.000	0.000	0.000	0
SN023175	0.000	0.000	0.000	4
SN023176	0.000	0.000	0.000	5
SN023177	0.200	0.167	0.182	6
SN023178	0.000	0.000	0.000	3
SN023179	0.000	0.000	0.000	5
SN023180	0.000	0.000	0.000	7

Continuation of Table 33				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN023181	0.000	0.000	0.000	4
SN023182	0.000	0.000	0.000	1
SN023183	0.000	0.000	0.000	1
SN023185	0.000	0.000	0.000	8
SN023186	0.000	0.000	0.000	2
SN023522	0.000	0.000	0.000	1
SN023523	0.000	0.000	0.000	12
SN023524	0.082	0.778	0.149	9
SN023527	0.000	0.000	0.000	3
SN023528	0.000	0.000	0.000	1
SN023531	0.000	0.000	0.000	10
SN023535	0.000	0.000	0.000	4
SN023538	0.000	0.000	0.000	6
SN023729	0.148	0.659	0.242	223
SN023730	0.785	0.508	0.617	6082
SN023731	0.363	0.276	0.314	1737
SN024400	0.000	0.000	0.000	6
SN024401	0.000	0.000	0.000	9
SN024402	0.000	0.000	0.000	8
SN024794	0.000	0.000	0.000	4
SN024795	0.000	0.000	0.000	7
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	9
SN024798	0.000	0.000	0.000	3
SN024799	0.000	0.000	0.000	11
SN024800	0.000	0.000	0.000	9
SN024801	0.000	0.000	0.000	6
SN024822	0.000	0.000	0.000	0
SN024823	0.000	0.000	0.000	2
SN024824	0.000	0.000	0.000	5
SN024825	0.000	0.000	0.000	6
SN024980	0.125	0.333	0.182	3
SN024981	0.038	0.500	0.071	2
SN024984	0.006	0.500	0.013	2

Continuation of Table 33				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN025079	0.000	0.000	0.000	3
SN025080	0.000	0.000	0.000	13
SN025081	0.000	0.000	0.000	6
SN025082	0.000	0.000	0.000	7
SN025083	0.000	0.000	0.000	14
SN025084	0.357	0.417	0.385	12
SN025331	0.407	0.658	0.503	1012
SN025332	0.666	0.816	0.734	7603
SN025333	0.063	0.012	0.020	1004
SN025334	0.075	0.418	0.127	170
SN025335	0.095	0.008	0.014	530
SN025336	0.518	0.215	0.304	5465
SN025337	0.695	0.919	0.791	7572
SN026227	0.000	0.000	0.000	13
SN026228	0.200	0.250	0.222	4
accuracy			0.587	32085
macro avg	0.043	0.071	0.044	32085
weighted avg	0.597	0.587	0.566	32085
End of Table				

Table 34. Precision, Recall, Accuracy, and F1-Scores for Decision Tree Classifier trained on both datasets.

Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN014855	0.667	0.222	0.333	9
SN014856	0.143	0.250	0.182	4
SN014857	0.385	1.000	0.556	5
SN016837	0.000	0.000	0.000	1
SN016838	0.000	0.000	0.000	8
SN016839	0.000	0.000	0.000	8
SN016840	0.000	0.000	0.000	1
SN016841	0.000	0.000	0.000	9
SN016842	0.100	0.167	0.125	6

Continuation of Table 34				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN016843	0.000	0.000	0.000	3
SN016844	0.273	0.333	0.300	9
SN016845	0.053	0.143	0.077	7
SN016846	0.000	0.000	0.000	4
SN016847	0.000	0.000	0.000	9
SN016848	0.333	0.200	0.250	10
SN016849	0.111	0.167	0.133	6
SN016850	0.000	0.000	0.000	4
SN017952	0.429	0.273	0.333	11
SN017953	0.125	0.250	0.167	4
SN017955	0.000	0.000	0.000	6
SN017956	0.000	0.000	0.000	7
SN017959	0.400	0.400	0.400	5
SN018064	0.000	0.000	0.000	7
SN018065	0.500	0.667	0.571	3
SN018066	0.000	0.000	0.000	1
SN018067	0.000	0.000	0.000	0
SN018068	0.200	0.167	0.182	6
SN018069	0.000	0.000	0.000	3
SN018070	0.000	0.000	0.000	2
SN018071	0.000	0.000	0.000	2
SN018072	0.167	0.500	0.250	2
SN018800	0.000	0.000	0.000	1
SN018802	0.333	0.154	0.211	13
SN018803	0.364	0.800	0.500	5
SN018804	0.143	0.200	0.167	5
SN018805	0.000	0.000	0.000	5
SN018806	0.500	0.182	0.267	11
SN018807	0.182	0.286	0.222	7
SN018808	0.000	0.000	0.000	5
SN018809	0.400	0.286	0.333	7
SN018810	0.000	0.000	0.000	6
SN018811	0.667	0.444	0.533	9
SN018812	0.143	0.200	0.167	5

Continuation of Table 34				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN019683	0.400	0.222	0.286	9
SN019684	0.500	0.100	0.167	10
SN019685	0.500	0.333	0.400	6
SN019688	0.500	0.357	0.417	14
SN020087	0.000	0.000	0.000	2
SN020089	0.000	0.000	0.000	2
SN020090	0.000	0.000	0.000	2
SN020091	0.000	0.000	0.000	1
SN020092	0.000	0.000	0.000	2
SN020093	0.000	0.000	0.000	3
SN020101	0.000	0.000	0.000	2
SN020102	0.143	0.111	0.125	9
SN020104	0.000	0.000	0.000	5
SN020105	0.143	0.200	0.167	5
SN020107	0.111	0.200	0.143	5
SN020109	0.000	0.000	0.000	4
SN020885	0.000	0.000	0.000	1
SN020886	0.000	0.000	0.000	0
SN020887	0.000	0.000	0.000	1
SN020888	0.000	0.000	0.000	1
SN020889	0.000	0.000	0.000	8
SN020890	0.286	0.167	0.211	12
SN020891	0.000	0.000	0.000	3
SN020892	0.200	0.167	0.182	6
SN020893	0.500	0.400	0.444	5
SN022017	0.000	0.000	0.000	5
SN022018	0.143	0.125	0.133	8
SN022312	0.000	0.000	0.000	0
SN022313	0.000	0.000	0.000	5
SN022314	0.000	0.000	0.000	2
SN022315	0.000	0.000	0.000	2
SN022316	0.333	0.125	0.182	8
SN022456	0.000	0.000	0.000	1
SN022457	0.000	0.000	0.000	1

Continuation of Table 34				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN022458	0.000	0.000	0.000	5
SN022459	0.125	0.125	0.125	8
SN022460	0.000	0.000	0.000	2
SN022464	0.000	0.000	0.000	3
SN022817	0.467	0.700	0.560	10
SN022818	0.143	0.667	0.235	3
SN022819	0.000	0.000	0.000	0
SN022823	0.000	0.000	0.000	3
SN023170	0.000	0.000	0.000	5
SN023172	0.000	0.000	0.000	0
SN023174	0.000	0.000	0.000	0
SN023175	0.000	0.000	0.000	4
SN023176	0.000	0.000	0.000	5
SN023177	0.333	0.500	0.400	6
SN023178	0.000	0.000	0.000	3
SN023179	0.000	0.000	0.000	5
SN023180	0.000	0.000	0.000	7
SN023181	0.250	0.500	0.333	4
SN023182	0.000	0.000	0.000	1
SN023183	0.500	1.000	0.667	1
SN023185	0.333	0.250	0.286	8
SN023186	0.000	0.000	0.000	2
SN023522	0.000	0.000	0.000	1
SN023523	0.250	0.083	0.125	12
SN023524	0.200	0.444	0.276	9
SN023527	0.167	0.333	0.222	3
SN023528	0.000	0.000	0.000	1
SN023531	0.200	0.200	0.200	10
SN023535	0.000	0.000	0.000	4
SN023538	0.133	0.333	0.190	6
SN023729	0.306	0.336	0.321	223
SN023730	0.811	0.779	0.795	6082
SN023731	0.477	0.512	0.494	1737
SN024399	0.000	0.000	0.000	0

Continuation of Table 34				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN024400	0.000	0.000	0.000	6
SN024401	0.000	0.000	0.000	9
SN024402	0.000	0.000	0.000	8
SN024794	0.000	0.000	0.000	4
SN024795	0.000	0.000	0.000	7
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	9
SN024798	0.000	0.000	0.000	3
SN024799	0.000	0.000	0.000	11
SN024800	0.250	0.111	0.154	9
SN024801	0.000	0.000	0.000	6
SN024823	0.200	0.500	0.286	2
SN024824	0.200	0.200	0.200	5
SN024825	1.000	0.167	0.286	6
SN024980	0.000	0.000	0.000	3
SN024981	1.000	0.500	0.667	2
SN024984	0.333	0.500	0.400	2
SN025079	0.000	0.000	0.000	3
SN025080	0.769	0.769	0.769	13
SN025081	0.125	0.167	0.143	6
SN025082	0.250	0.286	0.267	7
SN025083	0.412	0.500	0.452	14
SN025084	0.429	0.250	0.316	12
SN025331	0.549	0.513	0.530	1012
SN025332	0.745	0.746	0.746	7603
SN025333	0.258	0.296	0.275	1004
SN025334	0.083	0.100	0.090	170
SN025335	0.166	0.185	0.175	530
SN025336	0.544	0.545	0.545	5465
SN025337	0.858	0.843	0.850	7572
SN026227	0.200	0.385	0.263	13
SN026228	0.500	1.000	0.667	4
accuracy			0.680	32085
macro avg	0.167	0.176	0.157	32085

Continuation of Table 34				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
weighted avg	0.686	0.680	0.682	32085
End of Table				

Table 35. Precision, Recall, Accuracy, and F1-Scores for Random Forest Classifier trained on both datasets.

Sample ID	Precision	Recall	F1-Score	Support
SN014855	1.000	0.333	0.500	9
SN014856	0.333	0.500	0.400	4
SN014857	0.500	1.000	0.667	5
SN016837	0.000	0.000	0.000	1
SN016838	0.000	0.000	0.000	8
SN016839	0.000	0.000	0.000	8
SN016840	0.000	0.000	0.000	1
SN016841	0.000	0.000	0.000	9
SN016842	0.167	0.167	0.167	6
SN016843	0.000	0.000	0.000	3
SN016844	1.000	0.111	0.200	9
SN016845	0.000	0.000	0.000	7
SN016846	0.000	0.000	0.000	4
SN016847	0.167	0.111	0.133	9
SN016848	0.375	0.300	0.333	10
SN016849	0.250	0.167	0.200	6
SN016850	0.000	0.000	0.000	4
SN017952	0.667	0.545	0.600	11
SN017953	0.500	0.250	0.333	4
SN017955	0.000	0.000	0.000	6
SN017956	0.000	0.000	0.000	7
SN017959	0.500	0.600	0.545	5
SN018062	0.000	0.000	0.000	0
SN018064	0.000	0.000	0.000	7
SN018065	0.400	0.667	0.500	3
SN018066	0.000	0.000	0.000	1

Continuation of Table 35				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN018068	0.167	0.167	0.167	6
SN018069	0.000	0.000	0.000	3
SN018070	0.000	0.000	0.000	2
SN018071	0.000	0.000	0.000	2
SN018072	0.400	1.000	0.571	2
SN018800	0.000	0.000	0.000	1
SN018802	0.500	0.154	0.235	13
SN018803	0.500	0.800	0.615	5
SN018804	0.250	0.200	0.222	5
SN018805	0.000	0.000	0.000	5
SN018806	1.000	0.273	0.429	11
SN018807	1.000	0.143	0.250	7
SN018808	0.500	0.200	0.286	5
SN018809	0.222	0.286	0.250	7
SN018810	0.143	0.167	0.154	6
SN018811	1.000	0.333	0.500	9
SN018812	0.000	0.000	0.000	5
SN019683	0.333	0.111	0.167	9
SN019684	1.000	0.100	0.182	10
SN019685	0.250	0.167	0.200	6
SN019688	0.556	0.357	0.435	14
SN020087	0.000	0.000	0.000	2
SN020089	0.000	0.000	0.000	2
SN020090	0.000	0.000	0.000	2
SN020091	0.000	0.000	0.000	1
SN020092	0.000	0.000	0.000	2
SN020093	0.000	0.000	0.000	3
SN020100	0.000	0.000	0.000	0
SN020101	0.000	0.000	0.000	2
SN020102	0.400	0.222	0.286	9
SN020104	0.222	0.400	0.286	5
SN020105	0.167	0.200	0.182	5
SN020107	0.167	0.200	0.182	5
SN020109	0.200	0.250	0.222	4

Continuation of Table 35				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN020885	0.000	0.000	0.000	1
SN020886	0.000	0.000	0.000	0
SN020887	0.000	0.000	0.000	1
SN020888	0.000	0.000	0.000	1
SN020889	0.667	0.250	0.364	8
SN020890	0.400	0.167	0.235	12
SN020891	0.077	0.333	0.125	3
SN020892	0.500	0.167	0.250	6
SN020893	1.000	0.600	0.750	5
SN022017	0.000	0.000	0.000	5
SN022018	0.500	0.375	0.429	8
SN022312	0.000	0.000	0.000	0
SN022313	0.000	0.000	0.000	5
SN022314	0.000	0.000	0.000	2
SN022315	0.000	0.000	0.000	2
SN022316	0.200	0.125	0.154	8
SN022456	0.000	0.000	0.000	1
SN022457	0.000	0.000	0.000	1
SN022458	0.000	0.000	0.000	5
SN022459	0.444	0.500	0.471	8
SN022460	0.000	0.000	0.000	2
SN022464	0.000	0.000	0.000	3
SN022817	0.889	0.800	0.842	10
SN022818	0.143	0.667	0.235	3
SN022823	0.000	0.000	0.000	3
SN023170	0.000	0.000	0.000	5
SN023172	0.000	0.000	0.000	0
SN023175	0.250	0.250	0.250	4
SN023176	0.000	0.000	0.000	5
SN023177	0.500	0.667	0.571	6
SN023178	0.000	0.000	0.000	3
SN023179	0.000	0.000	0.000	5
SN023180	0.000	0.000	0.000	7
SN023181	0.286	0.500	0.364	4

Continuation of Table 35				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN023182	0.000	0.000	0.000	1
SN023183	0.333	1.000	0.500	1
SN023185	0.111	0.125	0.118	8
SN023186	0.000	0.000	0.000	2
SN023522	0.000	0.000	0.000	1
SN023523	0.400	0.167	0.235	12
SN023524	0.571	0.444	0.500	9
SN023527	0.000	0.000	0.000	3
SN023528	0.000	0.000	0.000	1
SN023531	0.250	0.200	0.222	10
SN023535	0.000	0.000	0.000	4
SN023538	0.000	0.000	0.000	6
SN023729	0.442	0.561	0.494	223
SN023730	0.816	0.915	0.863	6082
SN023731	0.730	0.508	0.599	1737
SN024399	0.000	0.000	0.000	0
SN024400	0.000	0.000	0.000	6
SN024401	0.000	0.000	0.000	9
SN024402	0.200	0.125	0.154	8
SN024794	0.000	0.000	0.000	4
SN024795	0.333	0.143	0.200	7
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	9
SN024798	0.000	0.000	0.000	3
SN024799	0.000	0.000	0.000	11
SN024800	0.375	0.333	0.353	9
SN024801	0.250	0.167	0.200	6
SN024823	0.200	0.500	0.286	2
SN024824	0.111	0.200	0.143	5
SN024825	0.000	0.000	0.000	6
SN024980	1.000	0.333	0.500	3
SN024981	0.250	0.500	0.333	2
SN024984	0.333	0.500	0.400	2
SN025079	0.000	0.000	0.000	3

Continuation of Table 35				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN025080	0.923	0.923	0.923	13
SN025081	0.500	0.500	0.500	6
SN025082	0.571	0.571	0.571	7
SN025083	0.636	0.500	0.560	14
SN025084	0.800	0.333	0.471	12
SN025331	0.580	0.699	0.634	1012
SN025332	0.796	0.843	0.819	7603
SN025333	0.427	0.375	0.400	1004
SN025334	0.148	0.159	0.153	170
SN025335	0.256	0.204	0.227	530
SN025336	0.705	0.575	0.633	5465
SN025337	0.875	0.932	0.903	7572
SN026227	0.778	0.538	0.636	13
SN026228	0.571	1.000	0.727	4
accuracy			0.765	32085
macro avg	0.253	0.217	0.209	32085
weighted avg	0.757	0.765	0.757	32085
End of Table				

Table 36. Precision, Recall, Accuracy, and F1-Scores for Extra Forest Classifier trained on both datasets.

Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN014855	0.800	0.444	0.571	9
SN014856	0.400	0.500	0.444	4
SN014857	0.500	1.000	0.667	5
SN016837	0.000	0.000	0.000	1
SN016838	0.333	0.125	0.182	8
SN016839	0.000	0.000	0.000	8
SN016840	0.000	0.000	0.000	1
SN016841	0.000	0.000	0.000	9
SN016842	0.400	0.333	0.364	6
SN016843	0.000	0.000	0.000	3

Continuation of Table 36				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN016844	1.000	0.111	0.200	9
SN016845	0.000	0.000	0.000	7
SN016846	0.000	0.000	0.000	4
SN016847	0.200	0.111	0.143	9
SN016848	0.333	0.100	0.154	10
SN016849	0.200	0.167	0.182	6
SN016850	1.000	0.250	0.400	4
SN017952	0.667	0.545	0.600	11
SN017953	1.000	0.250	0.400	4
SN017955	0.000	0.000	0.000	6
SN017956	0.000	0.000	0.000	7
SN017959	0.500	1.000	0.667	5
SN018064	0.273	0.429	0.333	7
SN018065	0.500	1.000	0.667	3
SN018066	0.000	0.000	0.000	1
SN018068	0.250	0.167	0.200	6
SN018069	0.000	0.000	0.000	3
SN018070	0.000	0.000	0.000	2
SN018071	0.000	0.000	0.000	2
SN018072	0.500	0.500	0.500	2
SN018800	0.000	0.000	0.000	1
SN018802	0.500	0.154	0.235	13
SN018803	0.308	0.800	0.444	5
SN018804	0.250	0.200	0.222	5
SN018805	0.000	0.000	0.000	5
SN018806	0.833	0.455	0.588	11
SN018807	0.750	0.429	0.545	7
SN018808	0.333	0.200	0.250	5
SN018809	0.500	0.571	0.533	7
SN018810	0.167	0.167	0.167	6
SN018811	1.000	0.333	0.500	9
SN018812	0.000	0.000	0.000	5
SN019683	1.000	0.222	0.364	9
SN019684	0.500	0.100	0.167	10

Continuation of Table 36				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN019685	0.000	0.000	0.000	6
SN019688	0.556	0.357	0.435	14
SN020087	0.000	0.000	0.000	2
SN020089	0.000	0.000	0.000	2
SN020090	0.000	0.000	0.000	2
SN020091	0.000	0.000	0.000	1
SN020092	0.000	0.000	0.000	2
SN020093	0.000	0.000	0.000	3
SN020100	0.000	0.000	0.000	0
SN020101	0.000	0.000	0.000	2
SN020102	1.000	0.222	0.364	9
SN020104	0.250	0.400	0.308	5
SN020105	0.222	0.400	0.286	5
SN020107	0.250	0.200	0.222	5
SN020109	0.000	0.000	0.000	4
SN020885	0.000	0.000	0.000	1
SN020886	0.000	0.000	0.000	0
SN020887	0.000	0.000	0.000	1
SN020888	0.000	0.000	0.000	1
SN020889	0.500	0.125	0.200	8
SN020890	0.400	0.167	0.235	12
SN020891	0.000	0.000	0.000	3
SN020892	0.200	0.167	0.182	6
SN020893	0.750	0.600	0.667	5
SN022017	0.000	0.000	0.000	5
SN022018	0.750	0.375	0.500	8
SN022312	0.000	0.000	0.000	0
SN022313	0.500	0.200	0.286	5
SN022314	0.000	0.000	0.000	2
SN022315	0.000	0.000	0.000	2
SN022316	0.200	0.125	0.154	8
SN022456	0.000	0.000	0.000	1
SN022457	0.000	0.000	0.000	1
SN022458	0.000	0.000	0.000	5

Continuation of Table 36				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN022459	0.308	0.500	0.381	8
SN022460	0.000	0.000	0.000	2
SN022464	0.000	0.000	0.000	3
SN022817	0.889	0.800	0.842	10
SN022818	0.133	0.667	0.222	3
SN022823	0.000	0.000	0.000	3
SN023170	0.000	0.000	0.000	5
SN023172	0.000	0.000	0.000	0
SN023174	0.000	0.000	0.000	0
SN023175	0.500	0.250	0.333	4
SN023176	0.000	0.000	0.000	5
SN023177	0.333	0.333	0.333	6
SN023178	0.000	0.000	0.000	3
SN023179	0.111	0.200	0.143	5
SN023180	0.000	0.000	0.000	7
SN023181	0.500	0.500	0.500	4
SN023182	0.000	0.000	0.000	1
SN023183	1.000	1.000	1.000	1
SN023185	0.333	0.250	0.286	8
SN023186	0.000	0.000	0.000	2
SN023522	0.000	0.000	0.000	1
SN023523	0.500	0.167	0.250	12
SN023524	0.429	0.333	0.375	9
SN023527	0.000	0.000	0.000	3
SN023528	0.000	0.000	0.000	1
SN023531	0.200	0.200	0.200	10
SN023535	0.000	0.000	0.000	4
SN023538	0.000	0.000	0.000	6
SN023729	0.495	0.439	0.466	223
SN023730	0.817	0.924	0.867	6082
SN023731	0.738	0.499	0.595	1737
SN024399	0.000	0.000	0.000	0
SN024400	0.000	0.000	0.000	6
SN024401	0.000	0.000	0.000	9

Continuation of Table 36				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN024402	0.500	0.125	0.200	8
SN024794	0.000	0.000	0.000	4
SN024795	0.500	0.143	0.222	7
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	9
SN024798	0.125	0.333	0.182	3
SN024799	0.000	0.000	0.000	11
SN024800	0.429	0.333	0.375	9
SN024801	0.143	0.167	0.154	6
SN024823	0.250	0.500	0.333	2
SN024824	0.000	0.000	0.000	5
SN024825	0.000	0.000	0.000	6
SN024980	1.000	0.333	0.500	3
SN024981	0.500	0.500	0.500	2
SN024984	1.000	0.500	0.667	2
SN025079	0.000	0.000	0.000	3
SN025080	0.786	0.846	0.815	13
SN025081	0.600	0.500	0.545	6
SN025082	0.500	0.571	0.533	7
SN025083	0.636	0.500	0.560	14
SN025084	0.571	0.333	0.421	12
SN025331	0.592	0.676	0.631	1012
SN025332	0.793	0.833	0.813	7603
SN025333	0.441	0.367	0.400	1004
SN025334	0.118	0.235	0.157	170
SN025335	0.304	0.234	0.264	530
SN025336	0.695	0.578	0.631	5465
SN025337	0.874	0.929	0.901	7572
SN026227	0.857	0.462	0.600	13
SN026228	0.667	1.000	0.800	4
accuracy			0.763	32085
macro avg	0.287	0.226	0.229	32085
weighted avg	0.757	0.763	0.756	32085
End of Table				

Table 37. Precision, Recall, Accuracy, and F1-Scores for Voting Classifier with all models trained on both datasets.

Sample ID	Precision	Recall	F1-Score	Support
SN014855	0.750	0.333	0.462	9
SN014856	0.222	0.500	0.308	4
SN014857	0.500	1.000	0.667	5
SN016837	0.000	0.000	0.000	1
SN016838	0.000	0.000	0.000	8
SN016839	0.000	0.000	0.000	8
SN016840	0.000	0.000	0.000	1
SN016841	0.000	0.000	0.000	9
SN016842	0.200	0.167	0.182	6
SN016843	0.000	0.000	0.000	3
SN016844	0.250	0.111	0.154	9
SN016845	0.071	0.143	0.095	7
SN016846	0.000	0.000	0.000	4
SN016847	0.000	0.000	0.000	9
SN016848	0.200	0.100	0.133	10
SN016849	0.167	0.167	0.167	6
SN016850	0.000	0.000	0.000	4
SN017952	0.556	0.455	0.500	11
SN017953	0.125	0.250	0.167	4
SN017955	0.000	0.000	0.000	6
SN017956	0.500	0.143	0.222	7
SN017959	0.500	0.400	0.444	5
SN018062	0.000	0.000	0.000	0
SN018064	0.000	0.000	0.000	7
SN018065	0.750	1.000	0.857	3
SN018066	0.000	0.000	0.000	1
SN018067	0.000	0.000	0.000	0
SN018068	0.250	0.167	0.200	6
SN018069	0.000	0.000	0.000	3
SN018070	0.000	0.000	0.000	2
SN018071	0.000	0.000	0.000	2
SN018072	0.500	0.500	0.500	2

Continuation of Table 37				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN018800	0.000	0.000	0.000	1
SN018802	0.400	0.154	0.222	13
SN018803	0.500	0.800	0.615	5
SN018804	0.200	0.200	0.200	5
SN018805	0.000	0.000	0.000	5
SN018806	0.400	0.364	0.381	11
SN018807	0.273	0.429	0.333	7
SN018808	0.000	0.000	0.000	5
SN018809	0.500	0.286	0.364	7
SN018810	0.125	0.167	0.143	6
SN018811	0.800	0.444	0.571	9
SN018812	0.500	0.200	0.286	5
SN019683	0.667	0.222	0.333	9
SN019684	1.000	0.100	0.182	10
SN019685	0.400	0.333	0.364	6
SN019688	0.750	0.429	0.545	14
SN020087	0.000	0.000	0.000	2
SN020089	0.000	0.000	0.000	2
SN020090	0.000	0.000	0.000	2
SN020091	0.000	0.000	0.000	1
SN020092	0.000	0.000	0.000	2
SN020093	0.000	0.000	0.000	3
SN020101	0.000	0.000	0.000	2
SN020102	0.200	0.111	0.143	9
SN020104	0.250	0.200	0.222	5
SN020105	0.000	0.000	0.000	5
SN020107	0.167	0.200	0.182	5
SN020109	0.000	0.000	0.000	4
SN020885	0.000	0.000	0.000	1
SN020886	0.000	0.000	0.000	0
SN020887	0.000	0.000	0.000	1
SN020888	0.000	0.000	0.000	1
SN020889	0.000	0.000	0.000	8
SN020890	0.333	0.167	0.222	12

Continuation of Table 37				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN020891	0.000	0.000	0.000	3
SN020892	0.250	0.167	0.200	6
SN020893	1.000	0.400	0.571	5
SN022017	0.000	0.000	0.000	5
SN022018	0.286	0.250	0.267	8
SN022312	0.000	0.000	0.000	0
SN022313	0.000	0.000	0.000	5
SN022314	0.000	0.000	0.000	2
SN022315	0.000	0.000	0.000	2
SN022316	0.250	0.125	0.167	8
SN022456	0.000	0.000	0.000	1
SN022457	0.000	0.000	0.000	1
SN022458	0.000	0.000	0.000	5
SN022459	0.059	0.125	0.080	8
SN022460	0.000	0.000	0.000	2
SN022464	0.000	0.000	0.000	3
SN022817	0.778	0.700	0.737	10
SN022818	0.143	0.667	0.235	3
SN022819	0.000	0.000	0.000	0
SN022823	0.000	0.000	0.000	3
SN023170	0.000	0.000	0.000	5
SN023172	0.000	0.000	0.000	0
SN023175	0.000	0.000	0.000	4
SN023176	0.000	0.000	0.000	5
SN023177	0.364	0.667	0.471	6
SN023178	0.000	0.000	0.000	3
SN023179	0.250	0.200	0.222	5
SN023180	0.000	0.000	0.000	7
SN023181	0.400	0.500	0.444	4
SN023182	0.000	0.000	0.000	1
SN023183	0.500	1.000	0.667	1
SN023185	0.400	0.250	0.308	8
SN023186	0.000	0.000	0.000	2
SN023522	0.000	0.000	0.000	1

Continuation of Table 37				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN023523	0.333	0.167	0.222	12
SN023524	0.294	0.556	0.385	9
SN023527	0.500	0.333	0.400	3
SN023528	0.000	0.000	0.000	1
SN023531	0.091	0.100	0.095	10
SN023535	0.000	0.000	0.000	4
SN023538	0.000	0.000	0.000	6
SN023729	0.337	0.610	0.435	223
SN023730	0.820	0.866	0.842	6082
SN023731	0.663	0.491	0.564	1737
SN024399	0.000	0.000	0.000	0
SN024400	0.000	0.000	0.000	6
SN024401	0.000	0.000	0.000	9
SN024402	0.000	0.000	0.000	8
SN024794	0.000	0.000	0.000	4
SN024795	0.000	0.000	0.000	7
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	9
SN024798	0.000	0.000	0.000	3
SN024799	0.000	0.000	0.000	11
SN024800	0.200	0.111	0.143	9
SN024801	0.100	0.167	0.125	6
SN024822	0.000	0.000	0.000	0
SN024823	1.000	0.500	0.667	2
SN024824	0.000	0.000	0.000	5
SN024825	1.000	0.167	0.286	6
SN024980	0.500	0.333	0.400	3
SN024981	0.143	0.500	0.222	2
SN024984	1.000	0.500	0.667	2
SN025079	0.000	0.000	0.000	3
SN025080	0.833	0.769	0.800	13
SN025081	0.500	0.500	0.500	6
SN025082	0.375	0.429	0.400	7
SN025083	0.778	0.500	0.609	14

Continuation of Table 37				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN025084	0.417	0.417	0.417	12
SN025331	0.514	0.722	0.601	1012
SN025332	0.775	0.844	0.808	7603
SN025333	0.412	0.262	0.320	1004
SN025334	0.100	0.229	0.139	170
SN025335	0.226	0.132	0.167	530
SN025336	0.694	0.518	0.594	5465
SN025337	0.859	0.935	0.896	7572
SN026227	0.778	0.538	0.636	13
SN026228	0.400	1.000	0.571	4
accuracy			0.743	32085
macro avg	0.224	0.198	0.192	32085
weighted avg	0.738	0.743	0.734	32085
End of Table				

Table 38. Precision, Recall, Accuracy, and F1-Scores for Voting Classifier with tree models only trained on both datasets.

Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN014855	1.000	0.444	0.615	9
SN014856	0.333	0.500	0.400	4
SN014857	0.500	1.000	0.667	5
SN016837	0.000	0.000	0.000	1
SN016838	0.000	0.000	0.000	8
SN016839	0.000	0.000	0.000	8
SN016840	0.000	0.000	0.000	1
SN016841	0.000	0.000	0.000	9
SN016842	0.500	0.333	0.400	6
SN016843	0.000	0.000	0.000	3
SN016844	1.000	0.111	0.200	9
SN016845	0.000	0.000	0.000	7
SN016846	0.000	0.000	0.000	4
SN016847	0.200	0.111	0.143	9

Continuation of Table 38				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN016848	0.600	0.300	0.400	10
SN016849	0.250	0.167	0.200	6
SN016850	0.000	0.000	0.000	4
SN017952	0.600	0.545	0.571	11
SN017953	0.500	0.250	0.333	4
SN017955	0.000	0.000	0.000	6
SN017956	0.000	0.000	0.000	7
SN017959	0.600	0.600	0.600	5
SN018064	0.300	0.429	0.353	7
SN018065	0.400	0.667	0.500	3
SN018066	0.000	0.000	0.000	1
SN018068	0.143	0.167	0.154	6
SN018069	0.000	0.000	0.000	3
SN018070	0.000	0.000	0.000	2
SN018071	0.000	0.000	0.000	2
SN018072	0.200	0.500	0.286	2
SN018800	0.000	0.000	0.000	1
SN018802	0.667	0.154	0.250	13
SN018803	0.444	0.800	0.571	5
SN018804	0.250	0.200	0.222	5
SN018805	0.000	0.000	0.000	5
SN018806	1.000	0.364	0.533	11
SN018807	0.667	0.286	0.400	7
SN018808	0.000	0.000	0.000	5
SN018809	0.429	0.429	0.429	7
SN018810	0.143	0.167	0.154	6
SN018811	1.000	0.333	0.500	9
SN018812	0.000	0.000	0.000	5
SN019683	0.500	0.111	0.182	9
SN019684	0.500	0.100	0.167	10
SN019685	0.333	0.333	0.333	6
SN019688	0.636	0.500	0.560	14
SN020087	0.000	0.000	0.000	2
SN020089	0.000	0.000	0.000	2

Continuation of Table 38				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN020090	0.000	0.000	0.000	2
SN020091	0.000	0.000	0.000	1
SN020092	0.000	0.000	0.000	2
SN020093	0.000	0.000	0.000	3
SN020100	0.000	0.000	0.000	0
SN020101	0.000	0.000	0.000	2
SN020102	0.600	0.333	0.429	9
SN020104	0.222	0.400	0.286	5
SN020105	0.143	0.200	0.167	5
SN020107	0.125	0.200	0.154	5
SN020109	0.200	0.250	0.222	4
SN020885	0.000	0.000	0.000	1
SN020886	0.000	0.000	0.000	0
SN020887	0.000	0.000	0.000	1
SN020888	0.000	0.000	0.000	1
SN020889	0.333	0.125	0.182	8
SN020890	0.333	0.167	0.222	12
SN020891	0.077	0.333	0.125	3
SN020892	0.167	0.167	0.167	6
SN020893	1.000	0.600	0.750	5
SN022017	0.333	0.400	0.364	5
SN022018	0.600	0.375	0.462	8
SN022313	0.000	0.000	0.000	5
SN022314	0.000	0.000	0.000	2
SN022315	0.000	0.000	0.000	2
SN022316	0.250	0.125	0.167	8
SN022456	0.000	0.000	0.000	1
SN022457	0.000	0.000	0.000	1
SN022458	0.000	0.000	0.000	5
SN022459	0.333	0.375	0.353	8
SN022460	0.000	0.000	0.000	2
SN022464	0.000	0.000	0.000	3
SN022817	0.889	0.800	0.842	10
SN022818	0.133	0.667	0.222	3

Continuation of Table 38				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN022823	0.000	0.000	0.000	3
SN023170	0.000	0.000	0.000	5
SN023172	0.000	0.000	0.000	0
SN023175	0.333	0.250	0.286	4
SN023176	0.000	0.000	0.000	5
SN023177	0.600	0.500	0.545	6
SN023178	0.000	0.000	0.000	3
SN023179	0.000	0.000	0.000	5
SN023180	0.000	0.000	0.000	7
SN023181	0.400	0.500	0.444	4
SN023182	0.000	0.000	0.000	1
SN023183	0.500	1.000	0.667	1
SN023185	0.111	0.125	0.118	8
SN023186	0.000	0.000	0.000	2
SN023522	0.000	0.000	0.000	1
SN023523	0.600	0.250	0.353	12
SN023524	0.800	0.444	0.571	9
SN023527	0.000	0.000	0.000	3
SN023528	0.000	0.000	0.000	1
SN023531	0.222	0.200	0.211	10
SN023535	0.000	0.000	0.000	4
SN023538	0.000	0.000	0.000	6
SN023729	0.491	0.516	0.503	223
SN023730	0.816	0.924	0.866	6082
SN023731	0.744	0.503	0.600	1737
SN024399	0.000	0.000	0.000	0
SN024400	0.000	0.000	0.000	6
SN024401	0.000	0.000	0.000	9
SN024402	0.500	0.125	0.200	8
SN024794	0.000	0.000	0.000	4
SN024795	0.000	0.000	0.000	7
SN024796	0.000	0.000	0.000	1
SN024797	0.000	0.000	0.000	9
SN024798	0.111	0.333	0.167	3

Continuation of Table 38				
Sample ID	**Precision**	**Recall**	**F1-Score**	**Support**
SN024799	0.000	0.000	0.000	11
SN024800	0.500	0.222	0.308	9
SN024801	0.200	0.167	0.182	6
SN024823	0.000	0.000	0.000	2
SN024824	0.125	0.200	0.154	5
SN024825	0.000	0.000	0.000	6
SN024980	0.000	0.000	0.000	3
SN024981	0.250	0.500	0.333	2
SN024984	1.000	0.500	0.667	2
SN025079	0.000	0.000	0.000	3
SN025080	0.917	0.846	0.880	13
SN025081	0.714	0.833	0.769	6
SN025082	0.571	0.571	0.571	7
SN025083	0.667	0.571	0.615	14
SN025084	0.667	0.333	0.444	12
SN025331	0.596	0.699	0.643	1012
SN025332	0.796	0.840	0.817	7603
SN025333	0.440	0.379	0.407	1004
SN025334	0.137	0.194	0.161	170
SN025335	0.284	0.215	0.245	530
SN025336	0.705	0.581	0.637	5465
SN025337	0.875	0.932	0.903	7572
SN026227	0.800	0.615	0.696	13
SN026228	0.571	1.000	0.727	4
accuracy			0.768	32085
macro avg	0.261	0.223	0.221	32085
weighted avg	0.759	0.768	0.759	32085
End of Table				

Appendix C. Siamese Network PRAF Scores

Table 39. Precision, Recall, Accuracy, and F1-Scores for binary cross-entropy with no moment transformation trained on TESCAN and tested on EProbe.

Sample ID	Precision	Recall	F1-Score	Precision	Recall	F1-Score
	Same			Not-Same		
SN014855	0.011752	0.552	0.023013	0.988808	0.460233	0.628114
SN014856	0.010968	0.8	0.021639	0.985775	0.161163	0.277034
SN014857	0.012083	1	0.023877	1	0.049256	0.093887
SN016838	0.011427	0.88	0.02256	0.987984	0.114733	0.20559
SN016839	0.011392	0.866	0.022488	0.987797	0.126128	0.223693
SN016841	0.009883	0.594	0.019442	0.984904	0.308	0.469254
SN016842	0.011735	0.929	0.023177	0.990937	0.090267	0.165462
SN016844	0.011137	0.83	0.021978	0.986368	0.143035	0.24984
SN016845	0.010251	0.654	0.020185	0.985086	0.265744	0.418571
SN016847	0.012166	0.994	0.024038	0.998867	0.061512	0.115887
SN016848	0.011022	0.784	0.021738	0.986389	0.182012	0.307316
SN016849	0.011744	0.924	0.023194	0.990869	0.095895	0.174867
SN016850	0.011014	0.805	0.02173	0.985981	0.159477	0.274547
SN017952	0.012256	1	0.024215	1	0.062884	0.118327
SN017953	0.012158	0.998	0.024023	0.999593	0.057116	0.108058
SN017955	0.011924	0.944	0.02355	0.992847	0.090384	0.165684
SN017956	0.012629	0.996	0.024942	0.999508	0.094558	0.172771
SN017959	0.012554	1	0.024797	1	0.085407	0.157373
SN018064	0.01217	0.98	0.024041	0.996911	0.075047	0.139585
SN018068	0.011716	0.925	0.023139	0.990682	0.092721	0.169571
SN018802	0.011376	0.842	0.022449	0.987834	0.149174	0.259206
SN018803	0.012382	1	0.024461	1	0.072512	0.135218
SN018804	0.011808	0.921	0.023316	0.991221	0.103721	0.187791
SN018805	0.012038	0.941	0.023772	0.99332	0.102012	0.185022
SN018806	0.010273	0.613	0.020207	0.985839	0.313267	0.475452
SN018807	0.010971	0.787	0.02164	0.986045	0.175	0.297246
SN018808	0.011578	0.903	0.022862	0.989227	0.10357	0.187508
SN018809	0.011929	0.959	0.023565	0.993796	0.076372	0.141844
SN018810	0.011221	0.838	0.022146	0.98685	0.14136	0.247297
SN018811	0.012706	0.962	0.025081	0.996634	0.130826	0.23129

Continuation of Table 39						
Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
SN018812	0.011073	0.801	0.021845	0.986429	0.168198	0.287392
SN019683	0.011829	0.946	0.023366	0.992315	0.081081	0.149913
SN019684	0.012068	0.965	0.023838	0.995027	0.08143	0.150541
SN019685	0.011488	0.869	0.022675	0.988461	0.130488	0.230542
SN019688	0.011723	0.909	0.023147	0.990379	0.108919	0.196254
SN020102	0.011691	0.933	0.023092	0.990685	0.08286	0.15293
SN020104	0.012356	1	0.024411	1	0.070581	0.131856
SN020105	0.011609	0.907	0.022924	0.989515	0.102058	0.185032
SN020107	0.011825	0.914	0.023347	0.991137	0.111826	0.200976
SN020109	0.011639	0.906	0.022984	0.989739	0.10543	0.190561
SN020889	0.012202	0.997	0.024109	0.999433	0.0615	0.11587
SN020890	0.012174	0.975	0.024047	0.996382	0.080058	0.148208
SN020891	0.01244	1	0.024574	1	0.076907	0.142829
SN020892	0.010992	0.81	0.021689	0.985723	0.152535	0.264188
SN020893	0.012108	1	0.023926	1	0.051256	0.097513
SN022017	0.012036	0.963	0.023775	0.994708	0.080872	0.149583
SN022018	0.012399	1	0.024495	1	0.073837	0.13752
SN022316	0.01218	0.978	0.024061	0.996719	0.077721	0.144198
SN022458	0.01191	0.938	0.023522	0.99248	0.095151	0.173654
SN022459	0.012027	0.948	0.023752	0.99364	0.094465	0.172528
SN022817	0.012103	1	0.023916	1	0.05086	0.096798
SN022818	0.012454	1	0.024601	1	0.07793	0.144592
SN023175	0.012248	0.993	0.024198	0.998819	0.068826	0.128777
SN023176	0.011661	0.919	0.023029	0.990109	0.094279	0.172164
SN023177	0.011929	0.951	0.023562	0.993266	0.084035	0.154959
SN023179	0.011612	0.907	0.022931	0.989541	0.102314	0.185453
SN023180	0.012146	1	0.024	1	0.054279	0.102969
SN023181	0.012438	1	0.024571	1	0.076779	0.142609
SN023183	0.012454	0.998	0.024602	0.999709	0.079837	0.147866
SN023185	0.011836	0.938	0.023376	0.991998	0.089372	0.163971
SN023523	0.011893	0.932	0.023487	0.992127	0.09964	0.181092
SN023524	0.012077	1	0.023865	1	0.048779	0.093021
SN023527	0.011506	0.877	0.022714	0.988588	0.123895	0.220195

Continuation of Table 39						
Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
SN023531	0.012295	1	0.02429	1	0.065849	0.123561
SN023535	0.012282	1	0.024265	1	0.06486	0.12182
SN023538	0.012239	0.998	0.024181	0.999633	0.063419	0.11927
SN024400	0.012391	1	0.024478	1	0.073186	0.13639
SN024401	0.011867	0.936	0.023438	0.992127	0.093779	0.171361
SN024402	0.01106	0.798	0.021818	0.986398	0.170337	0.290508
SN024795	0.012184	0.999	0.024075	0.9998	0.058233	0.110055
SN024797	0.012021	0.97	0.023749	0.995246	0.073035	0.136083
SN024798	0.011435	0.888	0.02258	0.988015	0.10736	0.193676
SN024799	0.011689	0.919	0.023085	0.990335	0.096512	0.175883
SN024800	0.012136	1	0.023981	1	0.053512	0.101587
SN024801	0.012244	1	0.024192	1	0.061942	0.116658
SN024824	0.012044	0.976	0.023795	0.995977	0.069081	0.129201
SN024825	0.012349	0.993	0.024394	0.998937	0.0765	0.142117
SN024980	0.010228	0.597	0.02011	0.985923	0.328198	0.492463
SN024981	0.016027	0.602	0.031222	0.99195	0.570233	0.724169
SN024984	0.012346	1	0.024391	1	0.069791	0.130475
SN025080	0.012299	0.995	0.024299	0.999181	0.070895	0.132397
SN025081	0.012277	0.826	0.024194	0.991176	0.227279	0.369769
SN025082	0.011031	0.796	0.02176	0.986252	0.170174	0.290265
SN025083	0.012002	0.956	0.023706	0.99401	0.084907	0.15645
SN025084	0.011195	0.832	0.022092	0.986751	0.145488	0.253587
SN026227	0.009875	0.634	0.019446	0.983944	0.260802	0.412317
SN026228	0.012435	1	0.024565	1	0.076535	0.142187
Accuracy			0.12651			0.12651
Micro Av	0.011853	0.91046	0.12651	0.991209	0.117394	0.12651
Macro Av	0.011843	0.91046	0.023377	0.993479	0.117394	0.201289
End of Table						

Table 40. Precision, Recall, Accuracy, and F1-Scores for binary cross-entropy with 4 moment transformation trained on TESCAN and tested on EProbe.

Sample ID	Precision	Recall	F1-Score	Precision	Recall	F1-Score
	Same			Not-Same		
SN014855	0.032309	1	0.062596	1	0.651733	0.78915
SN014856	0.648508	1	0.786782	1	0.993698	0.996839
SN014857	0.034953	1	0.067545	1	0.678953	0.808782
SN016838	0.033697	1	0.065198	1	0.666558	0.799922
SN016839	0.042622	1	0.081759	1	0.738814	0.849791
SN016841	0.03644	1	0.070319	1	0.692535	0.81834
SN016842	0.037226	1	0.07178	1	0.699267	0.823022
SN016844	0.060924	1	0.11485	1	0.820767	0.901562
SN016845	0.086896	1	0.159898	1	0.877814	0.934932
SN016847	0.065591	1	0.123107	1	0.834349	0.909695
SN016848	0.104855	1	0.189807	1	0.900733	0.947774
SN016849	0.045155	1	0.086408	1	0.754116	0.859825
SN016850	0.075369	1	0.140174	1	0.857349	0.923196
SN017952	0.050731	1	0.096562	1	0.782419	0.877929
SN017953	0.043157	1	0.082744	1	0.742198	0.852025
SN017955	0.049422	1	0.094189	1	0.776349	0.874095
SN017956	0.055897	1	0.105876	1	0.803605	0.89111
SN017959	0.038101	1	0.073405	1	0.706442	0.827971
SN018064	0.050421	1	0.096002	1	0.781012	0.877043
SN018068	0.054242	1	0.102902	1	0.797256	0.887192
SN018802	0.048183	1	0.091937	1	0.770302	0.870249
SN018803	0.039434	1	0.075875	1	0.716756	0.835012
SN018804	0.05591	1	0.105899	1	0.803651	0.891138
SN018805	0.043081	1	0.082604	1	0.741721	0.85171
SN018806	0.063103	1	0.118715	1	0.82736	0.905525
SN018807	0.048167	1	0.091908	1	0.770221	0.870198
SN018808	0.053362	1	0.101317	1	0.793721	0.884999
SN018809	0.035391	1	0.068362	1	0.68307	0.811695
SN018810	0.043797	0.746	0.082737	0.99637	0.810616	0.893946
SN018811	0.036566	1	0.070552	1	0.693628	0.819103
SN018812	0.042584	1	0.081689	1	0.73857	0.849629
SN019683	0.052359	1	0.099507	1	0.789547	0.882398

Continuation of Table 40						
Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
SN019684	0.038244	1	0.07367	1	0.707581	0.828753
SN019685	0.04281	1	0.082105	1	0.740012	0.850582
SN019688	0.04607	1	0.088082	1	0.759233	0.863141
SN020102	0.038234	1	0.073651	1	0.7075	0.828697
SN020104	0.045832	1	0.087646	1	0.757919	0.862291
SN020105	0.03257	1	0.063086	1	0.654616	0.791261
SN020107	0.051682	1	0.098285	1	0.78664	0.88058
SN020109	0.038497	1	0.07414	1	0.709581	0.830123
SN020889	0.037291	1	0.071901	1	0.699814	0.823401
SN020890	0.595221	0.822	0.690466	0.997921	0.9935	0.995706
SN020891	0.959322	0.849	0.900796	0.998247	0.999581	0.998914
SN020892	0.03733	1	0.071974	1	0.70014	0.823626
SN020893	0.059698	1	0.11267	1	0.816849	0.899193
SN022017	0.03785	1	0.072939	1	0.704419	0.826579
SN022018	0.03657	1	0.070559	1	0.693663	0.819127
SN022316	0.07373	1	0.137334	1	0.853919	0.921204
SN022458	0.045809	1	0.087604	1	0.757791	0.862208
SN022459	0.042398	1	0.081347	1	0.737372	0.848836
SN022817	0.036082	1	0.06965	1	0.68936	0.81612
SN022818	0.049327	1	0.094016	1	0.775895	0.873808
SN023175	0.057541	1	0.10882	1	0.809547	0.894751
SN023176	0.050525	1	0.096191	1	0.781488	0.877343
SN023177	0.076858	1	0.142745	1	0.860337	0.924926
SN023179	0.053726	1	0.101973	1	0.795198	0.885917
SN023180	0.05542	1	0.10502	1	0.801814	0.890007
SN023181	0.051314	1	0.097618	1	0.785023	0.879566
SN023183	0.081867	1	0.151343	1	0.869593	0.930248
SN023185	0.061192	1	0.115327	1	0.821605	0.902067
SN023523	0.060547	1	0.114181	1	0.819581	0.900846
SN023524	0.052609	1	0.09996	1	0.790605	0.883059
SN023527	0.046094	1	0.088125	1	0.75936	0.863223
SN023531	0.055075	1	0.1044	1	0.8005	0.889197
SN023535	0.053824	1	0.10215	1	0.795593	0.886162

Continuation of Table 40						
Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
SN023538	0.046393	1	0.088672	1	0.760988	0.864274
SN024400	0.041404	1	0.079517	1	0.730791	0.844459
SN024401	0.041859	1	0.080354	1	0.733837	0.846489
SN024402	0.058156	1	0.10992	1	0.811686	0.896056
SN024795	0.053876	1	0.102244	1	0.795802	0.886292
SN024797	0.045354	1	0.086772	1	0.755244	0.860557
SN024798	0.057894	1	0.109451	1	0.810779	0.895503
SN024799	0.043651	1	0.083651	1	0.745244	0.854029
SN024800	0.11183	0.95	0.200105	0.999363	0.912267	0.953831
SN024801	0.055685	1	0.105496	1	0.802814	0.890623
SN024824	0.056351	1	0.106689	1	0.805279	0.892138
SN024825	0.046984	1	0.08975	1	0.76414	0.866303
SN024980	0.038196	1	0.073581	1	0.707198	0.82849
SN024981	0.038275	1	0.073727	1	0.707826	0.82892
SN024984	0.038405	1	0.07397	1	0.70886	0.829629
SN025080	0.057501	1	0.108749	1	0.809407	0.894665
SN025081	0.06776	1	0.12692	1	0.840023	0.913057
SN025082	0.031984	1	0.061985	1	0.64807	0.786459
SN025083	0.07277	1	0.135667	1	0.851837	0.919991
SN025084	0.069051	1	0.129182	1	0.843233	0.91495
SN026227	0.049407	1	0.094162	1	0.776279	0.874051
SN026228	0.038305	1	0.073784	1	0.70807	0.829088
Accuracy			0.777905			0.777905
Micro Av	0.048884	0.992724	0.777905	0.999891	0.775407	0.777905
Macro Av	0.074376	0.992724	0.120312	0.999907	0.775407	0.871691
End of Table						

Table 41. Precision, Recall, Accuracy, and F1-Scores for triplet loss with no moment transformation trained on TESCAN and tested on EProbe.

Sample ID	Precision	Recall	F1-Score	Precision	Recall	F1-Score
	Same			Not-Same		
SN014855	0.01379	0.699	0.027046	0.991711	0.418721	0.588827
SN014856	0.0113	0.703	0.022243	0.988019	0.284791	0.442138
SN014857	0.033722	1	0.065244	1	0.666814	0.800106
SN016838	0.017963	0.967	0.03527	0.999005	0.385267	0.556081
SN016839	0.011874	0.742	0.023373	0.989473	0.281988	0.438896
SN016841	0.011887	0.749	0.023402	0.989536	0.276012	0.431629
SN016842	0.012609	0.813	0.024833	0.991697	0.259721	0.411636
SN016844	0.01689	0.932	0.033178	0.997863	0.369198	0.538979
SN016845	0.011509	0.775	0.022682	0.988558	0.226035	0.36794
SN016847	0.017039	0.929	0.033464	0.997814	0.376826	0.547055
SN016848	0.012933	0.778	0.025443	0.99173	0.309558	0.471837
SN016849	0.01488	0.875	0.029263	0.995567	0.32643	0.491655
SN016850	0.014105	0.907	0.027778	0.995903	0.262837	0.415908
SN017952	0.018656	0.959	0.0366	0.998848	0.413419	0.584794
SN017953	0.015573	0.872	0.0306	0.995872	0.359058	0.527815
SN017955	0.016348	0.824	0.03206	0.995191	0.423488	0.594147
SN017956	0.015964	0.943	0.031396	0.997959	0.324081	0.489274
SN017959	0.015658	0.838	0.030741	0.995161	0.387419	0.557717
SN018064	0.019833	1	0.038894	1	0.425326	0.596812
SN018068	0.014819	0.844	0.029128	0.994808	0.347581	0.515166
SN018802	0.019079	0.856	0.037327	0.996582	0.488267	0.655418
SN018803	0.026632	1	0.051882	1	0.575012	0.730168
SN018804	0.024344	0.96	0.047485	0.999159	0.552628	0.711648
SN018805	0.016994	0.897	0.033357	0.99699	0.396686	0.567552
SN018806	0.014182	0.77	0.027851	0.992967	0.377616	0.547155
SN018807	0.013957	0.848	0.027462	0.994208	0.30336	0.464875
SN018808	0.010428	0.689	0.020546	0.985142	0.239767	0.385669
SN018809	0.013757	0.772	0.027032	0.992617	0.356442	0.524529
SN018810	0.025019	0.962	0.048769	0.999217	0.564081	0.72109
SN018811	0.013562	0.705	0.026612	0.991576	0.403744	0.573837
SN018812	0.010025	0.649	0.019745	0.984234	0.254791	0.404792
SN019683	0.011114	0.692	0.021876	0.987548	0.284023	0.441165

Continuation of Table 41						
Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
SN019684	0.014208	0.803	0.027921	0.993537	0.35214	0.519982
SN019685	0.015004	0.88	0.029505	0.995767	0.328244	0.493734
SN019688	0.012354	0.727	0.024296	0.990303	0.324198	0.48848
SN020102	0.011261	0.708	0.02217	0.987899	0.277186	0.432907
SN020104	0.018448	0.978	0.036213	0.999353	0.39493	0.566133
SN020105	0.011622	0.718	0.022874	0.988818	0.289977	0.448445
SN020107	0.011869	0.743	0.023364	0.989467	0.280721	0.437359
SN020109	0.01102	0.715	0.021706	0.987116	0.253895	0.403903
SN020889	0.022597	0.93	0.044123	0.998473	0.532267	0.694376
SN020890	0.029817	0.989	0.057889	0.999796	0.625814	0.769786
SN020891	0.029051	0.989	0.056443	0.999792	0.61564	0.76204
SN020892	0.011748	0.735	0.023126	0.989154	0.281035	0.437709
SN020893	0.017228	0.937	0.033833	0.998068	0.378465	0.548819
SN022017	0.022498	0.915	0.043916	0.998165	0.537733	0.698934
SN022018	0.017963	0.833	0.035167	0.995889	0.470453	0.639031
SN022316	0.015113	0.867	0.029709	0.995512	0.343023	0.510235
SN022458	0.016137	0.9	0.031705	0.996798	0.36193	0.531043
SN022459	0.016612	0.909	0.032627	0.997181	0.374291	0.544285
SN022817	0.015751	0.895	0.030957	0.996521	0.349674	0.517693
SN022818	0.033334	1	0.064518	1	0.662802	0.797211
SN023175	0.018666	0.981	0.036635	0.999448	0.400291	0.571635
SN023176	0.015975	0.891	0.031388	0.996509	0.361826	0.530889
SN023177	0.019915	1	0.039053	1	0.427756	0.5992
SN023179	0.014107	0.827	0.027741	0.993904	0.327965	0.493189
SN023180	0.01686	0.916	0.03311	0.997429	0.378895	0.549175
SN023181	0.016922	0.848	0.033183	0.99588	0.427174	0.597889
SN023183	0.03175	1	0.061546	1	0.645395	0.784487
SN023185	0.018047	0.96	0.035429	0.998817	0.39264	0.56369
SN023523	0.011748	0.755	0.023136	0.989224	0.261512	0.413666
SN023524	0.020151	0.992	0.0395	0.999788	0.439116	0.610219
SN023527	0.014791	0.873	0.02909	0.995461	0.32386	0.488722
SN023531	0.015154	0.867	0.029788	0.995535	0.344826	0.512229
SN023535	0.017711	0.955	0.034777	0.99864	0.384105	0.554813

Continuation of Table 41						
Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
SN023538	0.017884	0.938	0.035099	0.998206	0.401035	0.572189
SN024400	0.018043	0.966	0.035424	0.998984	0.388674	0.559618
SN024401	0.0158	0.907	0.031059	0.996858	0.343047	0.510437
SN024402	0.012623	0.787	0.024848	0.991361	0.284209	0.441769
SN024795	0.017935	0.959	0.035211	0.998777	0.389384	0.560321
SN024797	0.017685	0.933	0.034712	0.998043	0.397395	0.568449
SN024798	0.017381	0.949	0.034137	0.998426	0.376151	0.546436
SN024799	0.013532	0.833	0.02663	0.993436	0.293872	0.453571
SN024800	0.016952	0.91	0.033284	0.997299	0.386384	0.556978
SN024801	0.017128	0.932	0.033638	0.997913	0.378116	0.548429
SN024824	0.014636	0.879	0.028793	0.995509	0.311895	0.474979
SN024825	0.01629	0.952	0.032032	0.998319	0.331523	0.497752
SN024980	0.033902	0.859	0.065229	0.997713	0.71536	0.833268
SN024981	0.211334	0.992	0.348437	0.999903	0.956953	0.977957
SN024984	0.017466	0.677	0.034053	0.993304	0.557151	0.713881
SN025080	0.018053	0.793	0.035302	0.995194	0.498442	0.664213
SN025081	0.012336	0.679	0.024231	0.989955	0.367849	0.536386
SN025082	0.011382	0.734	0.022416	0.988184	0.258674	0.410019
SN025083	0.011182	0.706	0.022015	0.98768	0.274058	0.429062
SN025084	0.012379	0.821	0.024391	0.991344	0.238384	0.384346
SN026227	0.012862	0.798	0.025316	0.991906	0.287837	0.446195
SN026228	0.013837	0.839	0.027225	0.993893	0.304698	0.466409
Accuracy			0.394061			0.394061
Micro Av	0.016095	0.860092	0.394061	0.995831	0.388642	0.394061
Macro Av	0.018994	0.860092	0.036465	0.995005	0.388642	0.548976
End of Table						

Table 42. Precision, Recall, Accuracy, and F1-Scores for triplet loss with 4 moment transformation trained on TESCAN and tested on EProbe.

Sample ID	Precision	Recall	F1-Score	Precision	Recall	F1-Score
	Same			Not-Same		
SN014855	0.023346	1	0.045627	1	0.513558	0.67861
SN014856	0.025702	1	0.050115	1	0.559209	0.717299
SN014857	0.022966	1	0.044901	1	0.505326	0.671384
SN016838	0.074272	1	0.138274	1	0.85507	0.921873
SN016839	0.023725	1	0.046351	1	0.521523	0.685528
SN016841	0.058896	1	0.111241	1	0.814198	0.897584
SN016842	0.022289	1	0.043606	1	0.489942	0.657666
SN016844	0.050705	1	0.096516	1	0.782302	0.877856
SN016845	0.027376	1	0.053292	1	0.586872	0.739659
SN016847	0.05952	1	0.112353	1	0.816267	0.898841
SN016848	0.023006	1	0.044977	1	0.506198	0.672153
SN016849	0.087207	1	0.160424	1	0.878291	0.935202
SN016850	0.022433	1	0.043882	1	0.493291	0.660676
SN017952	0.052645	1	0.100025	1	0.790756	0.883153
SN017953	0.021515	1	0.042124	1	0.471174	0.640542
SN017955	0.022252	1	0.043535	1	0.48907	0.65688
SN017956	0.074884	1	0.139334	1	0.856349	0.922616
SN017959	0.022386	1	0.043792	1	0.492209	0.659705
SN018064	0.051517	1	0.097986	1	0.785919	0.880128
SN018068	0.069633	1	0.1302	1	0.84464	0.915777
SN018802	0.022997	1	0.04496	1	0.506	0.671979
SN018803	0.020239	1	0.039675	1	0.437105	0.608313
SN018804	0.021899	1	0.042859	1	0.48064	0.649232
SN018805	0.027974	1	0.054426	1	0.595965	0.74684
SN018806	0.024305	1	0.047457	1	0.533221	0.695557
SN018807	0.047955	1	0.091521	1	0.769151	0.869514
SN018808	0.023967	1	0.046812	1	0.526465	0.689783
SN018809	0.021805	1	0.042679	1	0.47836	0.64715
SN018810	0.021597	1	0.042282	1	0.473233	0.642441
SN018811	0.021064	1	0.041259	1	0.459605	0.629766
SN018812	0.022594	1	0.044189	1	0.496977	0.663974
SN019683	0.023169	1	0.045288	1	0.509744	0.675272

Continuation of Table 42						
Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
SN019684	0.021294	1	0.041701	1	0.46557	0.635343
SN019685	0.057359	1	0.108495	1	0.808907	0.89436
SN019688	0.021712	1	0.042501	1	0.47607	0.64505
SN020102	0.021586	1	0.04226	1	0.472953	0.642184
SN020104	0.043731	1	0.083798	1	0.745733	0.854349
SN020105	0.023014	1	0.044993	1	0.506384	0.672317
SN020107	0.024485	1	0.0478	1	0.536733	0.698537
SN020109	0.024032	1	0.046935	1	0.527767	0.6909
SN020889	0.022998	1	0.044962	1	0.506023	0.671999
SN020890	0.023318	1	0.045574	1	0.512965	0.678092
SN020891	0.020804	1	0.040761	1	0.452709	0.623262
SN020892	0.023313	1	0.045564	1	0.51286	0.678001
SN020893	0.050279	1	0.095744	1	0.78036	0.876632
SN022017	0.023726	1	0.046352	1	0.521535	0.685538
SN022018	0.024218	1	0.047291	1	0.5315	0.694091
SN022316	0.029338	1	0.057003	1	0.615279	0.761824
SN022458	0.04686	1	0.089526	1	0.763488	0.865884
SN022459	0.048522	1	0.092554	1	0.771988	0.871324
SN022817	0.025661	1	0.050039	1	0.5585	0.716715
SN022818	0.021387	1	0.041879	1	0.467942	0.637548
SN023175	0.049903	1	0.095062	1	0.778616	0.87553
SN023176	0.049283	1	0.093936	1	0.775686	0.873675
SN023177	0.047578	1	0.090835	1	0.767233	0.868287
SN023179	0.044998	1	0.086122	1	0.753221	0.859242
SN023180	0.073915	1	0.137656	1	0.854314	0.921434
SN023181	0.213584	1	0.351989	1	0.957186	0.978125
SN023183	0.039311	1	0.075649	1	0.715837	0.834388
SN023185	0.054107	1	0.102659	1	0.796721	0.886861
SN023523	0.110278	1	0.198649	1	0.906186	0.950784
SN023524	0.091083	1	0.166959	1	0.883965	0.938409
SN023527	0.040823	1	0.078444	1	0.726791	0.841782
SN023531	0.039407	1	0.075827	1	0.716558	0.834878
SN023535	0.050033	1	0.095297	1	0.779221	0.875913

Continuation of Table 42						
Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
SN023538	0.044528	1	0.085259	1	0.750488	0.857462
SN024400	0.071434	1	0.133342	1	0.848849	0.918246
SN024401	0.043844	1	0.084005	1	0.746419	0.854799
SN024402	0.048272	1	0.092098	1	0.770744	0.870531
SN024795	0.039828	1	0.076605	1	0.719674	0.836989
SN024797	0.041569	0.797	0.079017	0.997007	0.786326	0.879221
SN024798	0.040042	1	0.077	1	0.721233	0.838042
SN024799	0.105507	1	0.190876	1	0.901419	0.948154
SN024800	0.032289	1	0.062559	1	0.651512	0.788988
SN024801	0.063155	1	0.118807	1	0.827512	0.905616
SN024824	0.200602	1	0.334169	1	0.953663	0.976282
SN024825	0.097305	1	0.177352	1	0.892128	0.942989
SN024980	0.02037	1	0.039927	1	0.440802	0.611885
SN024981	0.021586	1	0.042259	1	0.472942	0.642173
SN024984	0.020178	1	0.039559	1	0.435372	0.606633
SN025080	0.022414	1	0.043846	1	0.49286	0.66029
SN025081	0.039049	1	0.075163	1	0.713849	0.833036
SN025082	0.020389	1	0.039964	1	0.441337	0.6124
SN025083	0.022533	1	0.044073	1	0.495593	0.662738
SN025084	0.030641	1	0.05946	1	0.63214	0.774615
SN026227	0.178253	1	0.302572	1	0.946395	0.97246
SN026228	0.020113	1	0.039433	1	0.4335	0.604813
			0.651612			0.651612
Micro Av	0.031869	0.997667	0.651612	0.999958	0.647588	0.651612
Macro Av	0.044387	0.997667	0.082921	0.999966	0.647588	0.77465
End of Table						

Table 43. Precision, Recall, Accuracy, and F1-Scores for binary cross-entropy with no moment transformation trained on EProbe and tested on TESCAN.

Sample ID	Precision	Recall	F1-Score	Precision	Recall	F1-Score
	Same			Not-Same		
SN025331	0.110954	0.471	0.1796	0.90808	0.580667	0.70837
SN025332	0.152655	0.759	0.254186	0.952068	0.531889	0.682492
SN025333	0.123423	0.45	0.193715	0.91344	0.644889	0.756024
SN025334	0.102429	0.388	0.162072	0.901481	0.622222	0.736261
SN025335	0.100481	0.397	0.160372	0.900314	0.605111	0.723769
SN025336	0.129425	0.596	0.212667	0.925116	0.554556	0.693435
SN025337	0.222688	0.691	0.336827	0.955198	0.732	0.828836
SN023729	0.155455	0.721	0.255764	0.947967	0.564778	0.70784
SN023730	0.089247	0.288	0.136267	0.894877	0.673444	0.768528
SN023731	0.111543	0.316	0.164884	0.904563	0.720333	0.802004
Accuracy			0.61146			0.61146
Micro Av	0.130153	0.5077	0.61146	0.919284	0.622989	0.61146
Macro Av	0.12983	0.5077	0.205635	0.92031	0.622989	0.740756
End of Table						

Table 44. Precision, Recall, Accuracy, and F1-Scores for binary cross-entropy with 4 moment transformation trained on EProbe and tested on TESCAN.

Sample ID	Precision	Recall	F1-Score	Precision	Recall	F1-Score
	Same			Not-Same		
SN025331	0.186275	0.114	0.141439	0.905624	0.944667	0.924734
SN025332	0.146919	0.062	0.087201	0.902067	0.96	0.930132
SN025333	0.220183	0.24	0.229665	0.914703	0.905556	0.910106
SN025334	0.522739	1	0.686577	1	0.898556	0.946568
SN025335	0.681481	0.092	0.162115	0.907957	0.995222	0.949589
SN025336	0.130016	0.081	0.099815	0.901994	0.939778	0.920498
SN025337	0.720165	0.35	0.471063	0.93168	0.984889	0.957546
SN023729	0.283105	0.248	0.264392	0.91758	0.930222	0.923858
SN023730	0.359411	0.464	0.405063	0.938454	0.908111	0.923033
SN023731	0.1893	0.092	0.123822	0.904562	0.956222	0.929675
Accuracy			0.87552			0.87552

Continuation of Table 44						
Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
Micro Av	0.345727	0.2743	0.87552	0.921176	0.942322	0.87552
Macro Av	0.34396	0.2743	0.267115	0.922462	0.942322	0.931574
End of Table						

Table 45. Precision, Recall, Accuracy, and F1-Scores for triplet loss with no moment transformation trained on EProbe and tested on TESCAN.

Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
SN025331	0.120936	0.651	0.203979	0.92441	0.474222	0.626863
SN025332	0.291053	0.771	0.422582	0.968848	0.791333	0.871139
SN025333	0.131195	0.638	0.217636	0.929531	0.530556	0.675532
SN025334	0.110513	0.595	0.186404	0.912262	0.467889	0.618537
SN025335	0.107351	0.59	0.18165	0.90897	0.454889	0.606339
SN025336	0.149812	0.637	0.242574	0.936848	0.598333	0.730269
SN025337	0.205776	0.741	0.322104	0.959525	0.682222	0.797454
SN023729	0.180231	0.64	0.281257	0.944177	0.676556	0.788271
SN023730	0.101847	0.59	0.173708	0.902543	0.421889	0.574998
SN023731	0.119131	0.581	0.19772	0.918212	0.522667	0.666147
Accuracy			0.57019			0.57019
Micro Av	0.14033	0.6434	0.57019	0.934147	0.562056	0.57019
Macro Av	0.151784	0.6434	0.242961	0.930533	0.562056	0.695555
End of Table						

Table 46. Precision, Recall, Accuracy, and F1-Scores for triplet loss with 4 moment transformation trained on EProbe and tested on TESCAN.

Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
SN025331	0.0978	0.68	0.171005	0.894979	0.303	0.452727

Continuation of Table 46						
Sample ID	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
	Same			Not-Same		
SN025332	0.099485	0.657	0.172804	0.898999	0.339222	0.492578
SN025333	0.144444	0.845	0.246715	0.962651	0.443889	0.607605
SN025334	0.129066	0.746	0.220059	0.93981	0.440667	0.6
SN025335	0.204918	1	0.340136	1	0.568889	0.725212
SN025336	0.111345	0.687	0.191632	0.918277	0.390778	0.548246
SN025337	0.219292	0.632	0.325605	0.9483	0.75	0.837573
SN023729	0.171963	0.92	0.289764	0.982796	0.507778	0.669597
SN023730	0.104141	0.679	0.180585	0.907759	0.351	0.50625
SN023731	0.103955	0.623	0.178178	0.905915	0.403333	0.558161
			0.47956			0.47956
Micro Av	0.131076	0.7469	0.47956	0.941164	0.449856	0.47956
Macro Av	0.138641	0.7469	0.231648	0.935948	0.449856	0.599795
End of Table						

Bibliography

1. A. Kabata-Pendias and A. B. Mukherjee, *Trace elements from soil to human*. Springer Berlin Heidelberg, 2007.

2. T. L. Spano, A. Simonetti, E. Balboni, C. Dorais, and P. C. Burns, "Trace element and U isotope analysis of uraninite and ore concentrate: Applications for nuclear forensic investigations," *Applied Geochemistry*, vol. 84, pp. 277–285, sep 2017.

3. K. Lützenkirchen, M. Wallenius, Z. Varga, T. Wiss, A. Knott, A. Nicholl, and K. Mayer, "Nuclear forensics on uranium fuel pellets," *Radiochimica Acta*, 2019.

4. H. E. Frimmel, S. Schedel, and H. Brätz, "Uraninite chemistry as forensic tool for provenance analysis," *Applied Geochemistry*, vol. 48, pp. 104–121, 2014.

5. E. Keegan, M. J. Kristo, M. Colella, M. Robel, R. Williams, R. Lindvall, G. Eppich, S. Roberts, L. Borg, A. Gaffney, J. Plaue, H. Wong, J. Davis, E. Loi, M. Reinhard, and I. Hutcheon, "Nuclear forensic analysis of an unknown uranium ore concentrate sample seized in a criminal investigation in Australia," *Forensic Science International*, vol. 240, pp. 111–121, 2014.

6. M. Depine, H. Frimmel, P. Emsbo, A. Koenig, and M. Kern, "Trace element distribution in uraninite from mesoarchaean witwatersrand conglomerates (south africa) supports placer model and magmatogenic source." *Mineralium Deposita*, vol. 48, pp. 423–435, 2013.

7. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

8. M. Evaluation and I. Engineering, "Handbook of analytical methods for materials," 2001.

9. TESCAN, *Scanning Electron Microscope, MIRA 3 FEG-SEM, Instructions for Use*, 2011.

10. JEOL. (2020, November) Electron probe micro analyzer. [Online]. Available: https://www.jeol.co.jp/en/science/epma.html [Accessed: 2020-11-05]

11. JEOL, *Scientific/Metrology Instruments, Electron Probe Microanalyzer, JXA-8230*, 2020.

12. 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, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

13. G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R.* Springer Publishing Company, Incorporated, 2014.

14. R. H. Byrd, P. Lu, and J. Nocedal, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific and Statistical Computing*, vol. 16(5), pp. 1190–1208, 1995.

15. C. Zhu, R. H. Byrd, and J. Nocedal, "L-bfgs-b: Algorithm 778: L-bfgs-b, fortran routines for large scale bound constrained optimization," *ACM Transactions on Mathematical Software*, vol. 23(4), pp. 550–560, 1997.

16. A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems.* Sebastopol, CA: O'Reilly Media, 2019.

17. F. Chollet, *Deep Learning with Python*. Manning Publications, 2018.

18. R. Rodríguez-Pérez and J. Bajorath, "Interpretation of compound activity predictions from complex machine learning models using local approximations and shapley values," *Journal of Medicinal Chemistry*, vol. 63, no. 16, pp. 8761–8777, 2020, pMID: 31512867. [Online]. Available: https://doi.org/10.1021/acs.jmedchem.9b01101 [Accessed: 2020-12-15]

19. 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, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/ [Accessed: 2020-05-07]

20. G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," Tech. Rep., 2015. [Online]. Available: https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf [Accessed: 2020-05-24]

21. A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *CoRR*, vol. abs/1703.07737, 2017. [Online]. Available: http://arxiv.org/abs/1703.07737 [Accessed: 2020-07-10]

22. J. S. Milton and J. C. Arnold, *Introduction to Probability and Statistics 4th ed.* McGraw-Hill, 2003.

23. K. W. Bowyer, N. V. Chawla, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *CoRR*, vol. abs/1106.1813, 2011. [Online]. Available: http://arxiv.org/abs/1106.1813 [Accessed: 2020-10-21]

24. D. Gum, "A machine learning approach to characterizing particle morphology in nuclear forensics," 2020.

25. M. Benedict, T. H. Pigford, and H. W. Levi, *Nuclear Chemical Engineering. 2nd ed.* McGraw-Hill, 1981.

26. C. R. Pitkins, "Improving fallout characterization by using multivariate techniques to determine composition," 2018.

27. A. Holland, "Elemental analysis of actinide bearing particles," Air Force Office of Scientific Research, Tech. Rep., 2019.

28. McCrone Group, "Elemental assay measurement data," 2008-2019.

29. G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html [Accessed: 2020-10-21]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
21–03–2019	Master's Thesis	Sept 2019 — Mar 2021

4. TITLE AND SUBTITLE

Application of Artificial Neural Networks to Elemental Assay Data for Nuclear Forensics Analysis

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Seik, Jason G, Capt, USAF

5d. PROJECT NUMBER

XXXXXX

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering an Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENP-MS-21-M-135

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Technical Applications Center

10. SPONSOR/MONITOR'S ACRONYM(S)

AFTAC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

An Artificial Neural Network (ANN) is applied to elemental assay data of microscopic, actinide bearing particles obtained using energy dispersive x-ray spectroscopy via a scanning electron microscope (SEM-EDS) and Electron Probe Micro Analysis (EPMA). This technique provides a non-destructive assessment of the composition of particles that is suitable for nuclear forensics applications. A moment transformation was applied to the data before the ANN was used to compare and group like-particles together using a Siamese network and triplet loss function. A moment transformation provided a noticeable increase in accuracy across all runs. Models using triplet loss had nearly perfect precision when two observations were the same, and provided a preliminary means to compare unknown samples to a database of known samples. Adjusting the hyper parameters could further increase the performance of the models.

15. SUBJECT TERMS

Artificial Neural Networks, Siamese Network, Triplet Loss, Moment Transformation, Nuclear Forensics

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Abigail Bickley, AFIT/ENP
U	U	U	UU	185	19b. TELEPHONE NUMBER *(include area code)* (937) 255-6565; abigail.bickley@afit.edu