ABSTRACT

We propose a method for ensuring traceability of metal goods in an efficient and secure manner that leverages data obtained from micrographs of a part’s surface that is instance-specific (i.e., different for another instance of that same part). All stakeholders in modern supply chains face a growing need to ensure quality and trust in the goods they produce. Complex supply chains open many opportunities for counterfeiters, saboteurs, or other attackers to infiltrate supply networks, and existing methods for preventing such attacks can be costly, invasive, and ineffective. The proposed method extracts discriminatory-yet-robust intrinsic strings using features extracted from two-point autocorrelation data of surface microstructures. Using a synthetic dataset of three-phase micrographs similar to those obtained from metal alloy systems using low-cost optical microscopy techniques, we discuss the optimization of the method with respect to cost and security, and discuss the performance of the method in the context of anti-counterfeiting. Cryptographic extensions of this methodology are also discussed.

1 Introduction

Counterfeiting is a growing problem, especially in the context of globalized supply chain management [1,2]. Recent work has demonstrated the potential of using of microstructural features for anti-counterfeiting and traceability [3,4], leveraging the well-documented principles of physically unclonable functions (PUFs) [5,7]. In this paper, we present an extension of this work which (i) leverages a richer set of features derived from established work in materials characterization and homogenization [8,9], (ii) quantitatively evaluates the features in this space for information carrying capacity and resilience to expected damage, (iii) uses this quantification to select a subset of features most appropriate for traceability, and (iv) presents a method for combining the features in this subset for practical traceability in, for instance, supply chain management.

We begin with a summary of relevant literature in the area of computer vision and PUFs, and their relevance in anti-counterfeiting. We then introduce the motivation for using micrograph-derived features for traceability of metal parts and present the proposed anti-counterfeiting scheme. This scheme is evaluated using simulated micrograph data generated using the materials knowledge system python module [10,11].

1.1 Computer vision and physically unclonable functions in anti-counterfeiting

The computer vision literature covers a wide array of feature extraction algorithms, as well as accepted methods for analyzing the performance of classifiers built on the features extracted. For example, the scale-invariant feature transform (SIFT) is a well-known algorithm for extracting local feature vectors for object recognition and related computer vision tasks, and has been used for biometric applications like fingerprint matching [12]. Binary feature extraction algorithms like ORB and BRIEF have seen success recently, and are often faster to compute and require less storage than competing algorithms [13,14]. While such algorithms are effective in object recognition tasks, it is difficult to intuitively relate the features extracted to the structure of input images without clear target objects, as is often the case in random-field-like micrograph images [15].

Physically unclonable functions (PUFs) are functions applied to an input component, called the challenge, and provide an output response that marks the input as genuine or suspect [5,7].
PUF schemes have been used to address traceability in many domains, especially in electronics [16,17]. PUFs represent an active body of literature, and have inspired the approaches in this paper and previous papers by the authors [3,18], we refer the reader to [5] and [7] for comprehensive introductions to the field.

1.2 Use of statistical micrograph descriptors in materials science literature

Optical PUF schemes can be designed to leverage micrograph descriptors for determining authenticity [7]. For example, volume fraction and grain size descriptors have been studied for use in metal part anti-counterfeiting schemes in previous work by the authors [3,18], while paper fiber features have been used to develop schemes for packaging traceability [19]. The grain size and volume fraction descriptors have been shown to provide reasonably robust, small identifying strings for polygonal structures (360 brass microstructures) and multiphase alloys (4140-steel microstructures).

Grain boundary descriptors also make possible candidates for feature extraction, and have been used with success in materials characterization and evolution literature [20]. Geometric data or data on edges or vertices of grain boundaries, if they are expected to remain after damage, could serve as useful reproducible features for PUF design.

2-point auto-correlation and cross-correlation statistics are physically meaningful features that can be efficiently extracted from micrograph images where each pixel is assigned to one of \( N \) phases [21]. These statistics offer insight into the physically achievable structures of a given material system [8,9], and may be effectively compressed using dimensionality reduction techniques such as principal component analysis (PCA). These compressed representations have been used recently to establish highly accurate, computationally efficient structure-property linkages for materials characterization [22,23]. There is a demonstrated utility in the materials science literature for using reduced-dimension 2-point responses (using PCA) to train metamodels used to estimate the homogenized mechanical behavior of a microstructure instance [24]. Recent advancements such as the materials knowledge system (MKS) presented by Kalidindi, Latypov, and coauthors [11,25,26] formalize much of this work.

2 Leveraging Micrograph Data for Traceability

The goal of anti-counterfeiting schemes, PUF-based and otherwise, is to generate representations of part instances that are as unique to an instance as possible (discriminatory), but are insensitive to changes in the part’s structure caused by expected or acceptable damage (robust). For this paper, we consider such representations, referred to as strings, constructed from features that can be extracted from surface micrographs of a part. To meet the requirement of discriminatory ability, we desire features that are representative of the unique structure of a micrograph instance. To meet the robustness requirement, features should be insensitive to acceptable after-enrollment perturbations in the structure of the micrograph instance. As we shall see, these concerns motivate the feature set investigated in this paper.

Notably, the feature set discussed in this paper is built on two-point autocorrelation responses of image windows taken within the micrograph. We propose using PC scores, specifically those corresponding to the autocorrelations of each phase of the input micrograph, to build a high-dimensional feature response for each input micrograph, given a set of training micrographs used to generate the PC transform. From these scores, we propose an automated method for selecting “high-value” features for constructing the identifying string.

We pay particular attention to optimizing this process to produce discriminatory yet robust identifying strings, given expected damage types or imaging limitations when challenging a part’s origin. The experimental evaluation of our proposed anti-counterfeiting approach leverages the open-source materials knowledge system Python package [10] to generate micrograph libraries and the corresponding 2-point correlation data for the study presented in this paper, as well as software used for persistent homology computations [27] to extract high-value features from one-dimensional series data. In Section 5, we discuss refinements to this process that could enhance performance while controlling storage requirements in practical scenarios.

3 Application

In this section, we discuss the proposed anti-counterfeiting method. A summary of the key steps presented in this section is given in Figure [1] for reference. The method discussed was implemented and evaluated using a synthetic micrograph data set, with the data sampled using the PyMKS software package [10]. Three-phase micrographs were generated to simulate a multi-phase system as observed in alloys like steels, with two distinct “island” phases and one intermediate phase. The parameters used to generate this data set were chosen such that the resulting micrographs resembled 4140-steel micrographs gathered in lab for use in a previous study by the authors [3]. A visual comparison of this 4140-steel data and the generated data is provided in Figure [2]; note the pre-processed 4140-steel data has two “intermediate” phases while the synthetic data for this paper has one intermediate phase.

The parameters used for data generation are summarized in Table [1] and representative example micrographs are shown in Figure [3](A) and (B). As shown, the data took the form of two-dimensional arrays where each pixel was assigned one of three phases, \( N \in \{0,1,2\} \); these arrays may be thought of as microstructure function instances as described by Fullwood and coauthors [21], characterized by the parameters in Table [1] and discretized using a primitive basis with 3 states.

3.1 Damage profiles

Before discussing the implementation of the method, we first turn our attention to how the method will be evaluated in this paper. For evaluation and robustness testing, several damage profiles were simulated. Each profile was applied to the library of micrographs used for training and testing separately, and results for each of the profiles are compared in Section 4. Note that the profiles were selected either to (i) simulate poor challenge
imaging quality or loss of high-frequency phase data over time through Gaussian blurring of varying intensity (profiles 1-3), (ii) to simulate mis-aligned imaging or translation transformations of the microstructure when comparing the original “enrolled” response to the challenge (profiles 4-5), or (iii) to simulate mechanical or chemical damage taking the form of pitting (profiles 6-7). The profiles are summarized in Table 2, and those profiles involving blurring and pitting are illustrated in Figure 3 (C) through (F). Note that when blurring was applied, each pixel was assigned the phase closest to the resulting pixel value to ensure all locations in the sample were assigned to exactly one phase.

### 3.2 Feature extraction

Given the input micrograph libraries and damage profiles, we now look at how to extract the features and the subsequent strings from each instance in the micrograph libraries.

**PCA training for feature extraction:** Building on the previously-discussed motivation for using PCA responses in this study, 2,000 micrographs were initially generated to train the PCA models required. From each of these micrographs, a 200x200 pixel window was extracted. Then, the two-point autocorrelation statistic response, each of dimension 200x100 unique pixels, was calculated for each of these windows. These autocorrelations were calculated following the method of Fullwood and coauthors [21]. These responses were then used to train a PCA model for each of the three phases represented in the micrographs. As these involved large inputs, incremental PCA (IPCA) with a batch size of 50 micrograph windows was implemented using Python to train these models. An illustration of this pro-
TABLE 2. Damage profiles considered in this study. All units in pixels. For profiles 1 through 7, define \( \sigma_G = (\sigma_x, \sigma_y) \).

<table>
<thead>
<tr>
<th>Damage profile</th>
<th>Gaussian Blurring ( \sigma_x ) ( \sigma_y )</th>
<th>Translation in x in y</th>
<th>Pitting Pit count Pit radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (original)</td>
<td>0 0</td>
<td>0 0</td>
<td>0 N/A</td>
</tr>
<tr>
<td>1</td>
<td>4 8</td>
<td>0 0</td>
<td>0 N/A</td>
</tr>
<tr>
<td>2</td>
<td>4 4</td>
<td>0 0</td>
<td>0 N/A</td>
</tr>
<tr>
<td>3</td>
<td>8 8</td>
<td>0 0</td>
<td>0 N/A</td>
</tr>
<tr>
<td>4</td>
<td>2 2</td>
<td>20 20</td>
<td>0 N/A</td>
</tr>
<tr>
<td>5</td>
<td>2 2</td>
<td>40 40</td>
<td>0 N/A</td>
</tr>
<tr>
<td>6</td>
<td>2 2</td>
<td>0 0</td>
<td>15 25</td>
</tr>
<tr>
<td>7</td>
<td>2 2</td>
<td>0 0</td>
<td>20 40</td>
</tr>
</tbody>
</table>

FIGURE 2. (A) Pre-processed 4140-steel micrograph example used in [3]. (B) Example of a micrograph generated for use in this paper.

FIGURE 3. (A) Example three-phase generated micrograph. (B) 200 x 200 pixel window taken within the micrograph. (C) Damage profile 1, (D) Damage profile 2, (E) Damage profile 3, (F) Damage profile 7.

FIGURE 4. PCA transformation extraction pipeline. Incremental PCA parameters are given in Table 1. The cumulative explained variances for each of the first 20 PCA scores for each phase are plotted in Figure 5; note that near-complete information on the original micrographs is recovered after about 4 PC scores.

Sliding window feature extraction pipeline: Now, we leverage this PC transform. Note that the procedure for feature extraction is presented in Figure 7. From micrograph \( M \), we extract features from \( J \) windows \( m_{1 \leq j \leq J} \), where each window \( m_j \) is related in space to windows \( m_{j-1} \) and \( m_{j+1} \). In this paper, windows are taken sequentially along a circle of radius 150 pixels as in Figure 7(A). First, for each window, compute each phase’s 200x100-pixel autocorrelation response, and then apply the previously-computed PC transform for each of the three autocorrelations. Thus, each tile provides a set of PC scores, indexed by \( k \), for each of the three phases. If this is done for each tile along the path of the sliding window, then the result is a series-
of-series of PC scores for each phase as in Figure 7(B); these series may be used for further feature extraction explained below.

Let the PC score series corresponding to micrograph $M$, phase $\rho$, PC score $k$, be denoted $C^k_{M, \rho}$. It was observed during this study that common damage profiles, like scratching or pitting, can shift PC scores within the tiles affected an intolerable amount. But, the difference in PC responses between tiles was on the whole more robust. Define the series $\Delta C^k_{M, \rho}$ such that

$$\left(\Delta C^k_{M, \rho}\right)_n = (C^k_{M, \rho})_{n+1} - (C^k_{M, \rho})_n,$$  \hspace{1cm} (1)

then $\Delta C^k_{M, \rho}$ is the series of window-to-window differences in PC scores with a length one less than $C^k_{M, \rho}$, as in Figure 7(C). An illustration of PC responses for one phase is given in Figure 6(B) for a window moving at a rate of 5 pixels per step. Note the large difference in score between the damaged and undamaged micrographs, but the similar trajectories of the damaged and undamaged traces in Figure 6(B). The black rectangle in Figure 6(B) calls out a part of the path where a large difference in both absolute value and trajectory is observed. A robust feature extracted from this series of PC scores should be robust to these events if they occasionally occur in the data.

Now, we want to construct a set of features from this list of series, built on the de-correlated PC scores of the autocorrelation responses. There are many feature extraction methods that could be used here; in this paper, we take the five most dominant frequencies of the Fourier transform (FT) of each series after 0-padding, along with their relative persistence as discussed in literature on topological data analysis [29–30]. If each series $\Delta C^k_{M, \rho}$ has $N$ entries indexed by $n$, then the entries of the Fourier transform are

$$\left(\mathcal{F}_{M, \rho}^k\right)_n = \sum_{v=0}^{V} \left(\Delta C^k_{M, \rho}\right)_v e^{-2\pi i n v/V}.$$  \hspace{1cm} (2)

Since we only want to deal with real numbers, we consider the modulus of each entry, $|\mathcal{F}_n|$, for this analysis. The transform was computed using the Fast Fourier Transform (FFT) algorithm implemented in Python’s numpy library.

The peaks of $|\mathcal{F}|$ are found using principles in the literature on topological data analysis (TDA) [29, 30], particularly persistent homology. For each FT, the “barcode” of the signal is computed, which records the most significant peaks of the 1-D signal and a measure of their relative importance, called the persistence, in a noise-resilient way [28]. The locations and persistence scores of the first 5 peaks are recorded as features, such that each FFT supplies 9 features to the total feature list (the first peak’s relative persistence is always 1, and so is not useful). The peak and persistence results were calculated using the algorithm presented by Edelsbrunner [30] and implemented in the libstick software package [27]. A representative result is shown in Figure 7(D), with blue bars and their heights corresponding to the locations and order of significance of the persistent peaks, respectively. This method has the advantage of extracting noise-resilient features, at the expense of some correlation between the locations and persistences of the recorded peaks, as can be observed in Figure 8(A) discussed in the following sub-section.

3.3 Feature characterization

It is worth commenting briefly on how these features may be characterized in the context of anti-counterfeiting. As discussed above, features are constructed from the FT response of the PC responses along a “series” of smaller micrograph windows. Features correspond to (i) a phase $N$, (ii) a corresponding PC score for that phase, (iii) the location of the corresponding significant peak of the FT response for that PC score, and (iv) the relative significance of that response, if the peak is not the first-most-significant. In total, for this analysis 540 features are

FIGURE 5. Cumulative explained variance for each phase PCA transform for the first 20 PCA components.

FIGURE 6. (A) An example of a sliding window scanning down an image, comparing an undamaged micrograph (left) to a scratched version of the micrograph (right). (B) The corresponding first two PC scores, plotted for each window.
extracted from every input micrograph. Here, we discuss the desired properties of the extracted features and analyze a subset of these features for one damage profile for illustration. As discussed in Section 2, it is important to analyze the discriminatory ability and robustness of each feature, as this relates to the feature’s usefulness when included in the full micrograph string for anti-counterfeiting. Intuitively, features with low covariance with other features in the set should have higher information carrying potential, enhancing discriminatory ability while reducing the number of other feature responses that need to be stored. Also, features with low expected differences between response before and after a damage profile is applied are desired, as this indicates that feature is robust to that damage profile. It is also desired that the differences between one feature response before and after damage have low covariance with the differences of other features, as this implies that if one feature is changed due to damage, other features are not any more likely to change with it.

As an illustration, a feature covariance matrix containing data on 100 of the extracted features is plotted in Figure 8(A), along with the covariance matrix of the before-and-after feature differences for damage profile 2 in Figure 8(B). Note that the features were ordered using the value metrics discussed in the next section when considering exposure to damage profile 2. For this damage profile, the high-value features display particularly low covariance with other features; this may not always be the case if feature covariance is not explicitly considered when ordering features. In fact, for the analysis presented in the next subsection, low variance before and after damage was given priority when selecting features for robustness, while high variance between different micrograph instances was given priority for discriminatory ability. For all features reported, before-and-after feature differences in Figure 8(B) have low covariance relative to in-feature variance. This implies that feature differences due to damage should be fairly independent of each other, which is desired. With these characterizations in mind, let us continue and construct the output strings from these features.

### 3.4 Feature selection and bit string construction

From this feature set, we construct a representative string for an input micrograph as follows. For each feature $y_i$, take the histogram of responses from the enrolled data as discussed in the previous subsection. For the number of bins to be recorded, segment the histogram according to the estimated cumulative distribution. Then, assign the input micrograph string an integer corresponding to the matching bin for that feature. This generates strings of length (number of bins-1)*(number of features) bits, assuming each feature has the same number of bins. Let $B$ denote the number of bins minus 1, and let $N$ denote the number of features considered in the string. In this paper, we set $B = 7$ so each feature integer takes a value from 1 to 8. This scheme is illustrated in Algorithm 1.

**Computing distances between strings**: to evaluate this scheme, we must define a method to compute distances between the generated strings. This distance, then, captures our estimate of the dissimilarity between the input micrographs the strings represent. For this paper, we use a modified Hamming distance metric that compares each feature represented in the strings being compared, and adds to the distance based on how dissimilar those responses are to each other. If $s_1$ and $s_2$ are the strings being compared, and $(b_{i1})$ and $(b_{i2})$ are the corresponding integers in each string for feature $i$, then call this distance $D_{MH}(s_1, s_2)$ with

$$D_{MH}(s_1, s_2) = \frac{1}{BN} \sum_{i=1}^{N} ((b_{i1}) - (b_{i2})), \quad (3)$$

where $\frac{1}{BN}$ is a normalizing term. $D_{MH}(s_1, s_2) = 0$ implies $s_1 = s_2$, and so the input micrographs are nearly or even exactly identical, while $D_{MH}(s_1, s_2)$ close to 1 implies the features represented in the strings, and therefore the corresponding micrographs, are highly different from each other.

The distribution of distances over input strings computed us-
Data: Desired feature responses \( y_i \in Y \), cumulative distribution functions for each feature \( F_{i}(y_i) \), desired number of histogram bins for each feature \( B + 1 \).

Result: String \( s \)

1. Initialize \( s \) as an empty string
   \( s \leftarrow \{\} \)
2. Assign each \( y_i \) an integer identifier using corresponding CDF
   \( s \leftarrow s | b_i \)

for \( y_i \) in \( Y \) do
   for \( k \) in \( \{1, \ldots, B + 1\} \) do
     if \( \frac{k}{B} - 1 \leq F_i(y_i) < \frac{k}{B} \) then
       \( b_i \) is assigned \( k \).
       \( b_i \leftarrow k, \)
       break
   end
   Append \( b_i \) to string \( s \)
   \( s \leftarrow s | b_i \)
end

Return \( s \)

Algorithm 1: Construction of string \( s \) from features \( y_i \in Y \).

The final string is the concatenation of \( b_i \) for each desired feature for the input micrograph.

The “value” of each feature may be thought of as its contribution both to the discriminatory ability and the robustness to expected damage or information loss of the output string. Consider each feature response \( b_i \) in the string as having a discriminatory score, \( v_d(b_i) \), and a robustness score \( v_r(b_i) \). One natural way to compute \( v_d(b_i) \) would be to take the average normalized difference between string responses for feature \( y_i \) across micrographs in the data set corresponding to different part instances. So, let

\[
v_d(b_i) = \frac{1}{B[|N_{\text{diff}}|]} \sum_{(a,b) \in N_{\text{diff}}} ((b_i)_a - (b_i)_b), \tag{4}\]

where \( N_{\text{diff}} \) is the set of all pairs of strings taken from different micrographs and \( |N_{\text{diff}}| \) is that set’s length. This may be thought of as an estimate of the inter-distance of strings containing only this feature, which we want to maximize. Similarly, \( v_r(b_i) \) may be computed by taking the average normalized difference when considering strings corresponding to the same part instances before and after damage. So, let

\[
v_r(b_i) = 1 - \frac{1}{B[|N_{\text{same}}|]} \sum_{(a,b) \in N_{\text{same}}} ((b_i)_a - (b_i)_b), \tag{5}\]

where \( N_{\text{same}} \) is the set of all pairs of strings taken from the same micrograph before and after a specified damage profile. Note that here we subtract the distance average from 1: it is more valuable to minimize this distance average, as this average is an estimate of the intra-distance of the feature \( y_i \) and we want distances between feature strings before and after acceptable damage to be small. Since we want higher \( v_r(b_i) \) to be desired, subtracting this summation from 1 provides a correct score.

Now, for an input library of micrographs, we can compute each feature’s \( v_d(b_i) \) and \( v_r(b_i) \). A linear combination of these may be taken as a final value score for each feature. For this paper, we consider the average of the two scores,

\[
v(b_i) = \frac{1}{2} (v_d(b_i) + v_r(b_i)). \tag{6}\]

The ordered feature value scores for each damage profile are plotted in Figure 9, the three lines plotted for each profile are the results for three-fold cross validation. Responses are colored according to the corresponding damage profile. Scatter plots for feature values of \( v_d \) and \( v_r \) are provided in Figure 10 with points colored according to the phase autocorrelation that feature was computed from (see Figure 5 for phase colors). Note that the phases contributing the most valuable features (those maximizing both \( v_d \) and \( v_r \)) change between damage profiles, indicating that engineers must carefully anticipate which damage profiles are expected before committing to a set of features.
FIGURE 10. Scatter plots for feature value scores $v_d$ and $v_r$, for four damage profiles: (A) damage profile 3 (blurring), (B) damage profile 5 (translation), (C) damage profile 6 (moderate pitting), and (D) damage profile 7 (severe pitting).

4 String Construction and Results

Strings were constructed for each micrograph before and after applying each damage profile by taking only the highest-scoring features, up to a specified percent. By estimating the intra-distance and inter-distance distributions on one (testing) fold of the data, after using the other two folds to find value scores for each feature, a maximum-likelihood estimate classifier was constructed to label distances between before-and-after-damage pairs of micrograph strings $(s_1, s_2)$ as genuine (drawn from the intra-distance distribution) or counterfeit (drawn from the inter-distance distribution). If on comparison, it is more probable that $D_{MH}(s_1, s_2)$ was drawn from the intra-distance distribution than the inter-distance distribution assuming equal a priori probability, then $s_2$ is considered the genuine part that created $s_1$. Else, $s_2$ is labeled a counterfeit.

Estimated probabilities of mis-classification for strings constructed from $x$ input features, with order given by the feature value scores for that damage profile, are plotted in Figures 11 (A) (where strings are taken from only one “path” location in the micrograph) and 11 (B-D) (where strings are taken from two or four paths in the micrograph and concatenated to improve performance). $D_{MH}$ was computed for every $(s_1, s_2)$ combination in the test fold of the data, with $s_1$ coming from the original data for each micrograph and $s_2$ from the damaged micrographs. The estimated mis-classification probability for each damage profile is calculated as

$$P(\text{error}) = P(\text{CF}) \int_0^\tau p(D_{MH}|\text{CF})dD_{MH}$$

$$+P(\text{Genuine}) \int_\tau^1 p(D_{MH}|\text{Genuine})dD_{MH},$$

where $P(\text{CF}) = P(\text{Genuine}) = 0.5$ are the a priori probabilities that the part is a counterfeit or genuine, respectively, $\tau$ is the distance value $D_{MH}$ such that $p(\tau|\text{CF}) = p(\tau|\text{Genuine})$, and $p(D_{MH}|\text{CF})$ and $p(D_{MH}|\text{Genuine})$ are the normally-distributed probability density functions fitted to the damage profile’s inter- and intra-distance $D_{MH}$ histograms, respectively.

For most damage profiles, it can be seen that performance does not improve after around 250 included features. For profiles 3 and 7, representing more severe blurring and pitting respectively, performance drops off drastically once more than 250 features have been included in the string. In practice, one wants to select the number of features to consider that minimizes mis-classification probability, given the most likely damage profiles. Damage profiles corresponding to translation (4 and 5) are consistently poor performers over varying string length. This may be due to the fact that translation alters all pixel responses in the input tile window, which affects the autocorrelation and PC score responses much more drastically than the more localized blurring or pitting damage.

For more severe damage types with high mis-classification probability over all string lengths, reliability may be increased by performing feature extraction and string construction at two or more locations in the part’s micrograph. This increases string storage requirements given a desired number of features to extract per location, but results in better identifier performance as seen in Figure 11 (B-D). Results for strings considering one location are given in Figure 11 (A), while results for strings considering two and four locations in each micrograph are given in Figure 11 (B) and (C-D) respectively. Results for strings considering two or more locations were generated by creating groups of original instances in the original dataset, and taking each group as one instance in the new dataset.

Note that the less-severe damage profiles achieve good performance (less than 0.01 probability of misclassification) with only one string location, but more severe profiles require at least two locations to be considered for practically acceptable performance. In practice, this may be done by enforcing two or more responses per micrograph challenge input. For visualization, several inter-intra-distance plots are given in Figure 12 with the corresponding mis-classification probabilities. Note the significant inter-distance-intra-distance histogram overlap for more severe damage profiles considering only one location for feature extraction; this highlights the need to evaluate this probability before committing to an anti-counterfeiting scheme design.

An engineer using this method may proceed to design an anti-counterfeiting scheme from these results as follows:

1. The engineer provides training data for damage profiles for the parts under consideration, as described in Sections 3.1-3.2.
2. Knowing which data profiles are most likely to occur, perhaps through experience or experiment, the engineer identifies the most “valuable” features for those profiles using Equations 3-5.
3. Based on these results, the engineer estimates the per-part-
location performance given a set of ranked features using Figure 11.

4. The engineer selects the number and identity of features to extract, and the number of locations from which to extract features, that minimizes the expected probability of misclassification subject to string length constraints.

5. A Keyed Bitstring Computation Scheme

In practice, it may be desirable to build a part’s bitstring in a way such that a determined adversary would not be able to reliably replicate the structure that generated it, even with a large amount of resources dedicated to producing counterfeits. We propose two additional bitstring construction steps to address this: (i) select features to include in the final bitstring in a keyed fashion, such that even at challenge time, you do not know which features will contribute to the output string, and (ii) adaptively select the number of bits to assign per feature to best utilize each feature’s information carrying capacity and reduce the predictability of output string’s structure. An improved, keyed bitstring computation algorithm based on these ideas follows. Note that it is keyed, with the secret key used in 2 separate places (the subset-selection step, and the choice of random permutation).

1. Quality filtering: We narrow down the candidate features to the $n$ highest-value ones, where value is determined as in Section 3.4.
2. Subset selection: We then use the secret key to randomly select a fraction $c$ of those $n$ highest-value features, so we end up with a set of $m = cn$ selected features that will be used for encoding. For the sake of definiteness, we henceforth assume that $c = 0.5$. Note that an adversary that does not have the key faces a number of possible feature choices that, for $c = 0.5$, is about $2^{n/2}$ (a large number).

3. Bits per feature: We use the algorithm of the next section to compute the number of bits $b_i$ for each feature $y_i$, $1 \leq i \leq m$. The total bitstring $\sigma$ will therefore have length $\ell = \sum_{i=1}^{m} b_i$. The next steps explain how $\sigma$ is created.

4. Bitstring construction: After obtaining the bitstring $\sigma_i$ of length $b_i$ for each feature $y_i$, we create the overall bitstring $\sigma$ as follows.

(a) We use the secret key to select a random permutation $\pi$ of the $m$ features, uniformly from the $m!$ possible permutations.

(b) We create $\sigma$ by concatenating the $\sigma_i$ in the order determined by the permutation $\pi$. In other words, if $\|\|$ denotes concatenation, then

$$\sigma = \sigma_{\pi(1)}|\sigma_{\pi(1)}|\cdots|\sigma_{\pi(1)}$$ \hspace{1cm} (8)

5.1 Computing the number of bits per feature

The main idea underlying the computation of $b_1, \ldots, b_m$ is that a relatively higher-value feature should have somewhat more bits than a relatively lower-value feature (even though all $m$ features meet the quality-filtering criterion that selected them, such
bits approaches the number required to assign $\beta_{\text{min}}$ bits to each of the remaining $m - \alpha$ features. From that point on assign $\beta_{\text{min}}$ bits to to each of the remaining $m - \alpha$ features.”

To know what $\alpha$ is, we solve for it in the equation

$$\alpha \beta_{\text{max}} + (m - \alpha) \beta_{\text{min}} = \ell \quad (10)$$

which gives

$$\alpha = (\ell - m \beta_{\text{min}}) / (\beta_{\text{max}} - \beta_{\text{min}}) \quad (11)$$

As $\alpha$ must be integer and the right-hand side is typically not, we choose $\alpha$ to be the integer that is nearest to the right-hand side. This results in a length for $\sigma$ that can slightly differ from the target $\ell$ (by no more than $\beta_{\text{max}} - \beta_{\text{min}}$).

The above algorithm can easily be modified for the case where the $\beta_{\text{min}}$ and $\beta_{\text{max}}$ are feature-dependent, i.e., for feature $i$ they are $\beta_{\text{min}}^{(i)}$ and $\beta_{\text{max}}^{(i)}$. To simplify notation, we assume WLOG that $v_1 \geq v_2 \geq \ldots \geq v_m$, and we use $L_i$ to denote $\sum_{k=1}^{i} \beta_{\text{max}}^{(k)}$, $R_i$ to denote $\sum_{k=i}^{m} \beta_{\text{min}}^{(k)}$.

“Go through the features in the order 1, 2, $\ldots$, $m$, and keep assigning the maximum allowed $\beta_{\text{max}}$ bits to each feature $i$ encountered, while maintaining $L_i$ and $R_i$ as you go along. When the number of remaining bits (which is $\ell - L_i$) approaches the number required to assign the minimum bits to each of the remaining $m - i$ features, start assigning that minimum $\beta_{\text{min}}^{(i)}$ bits to each feature $j$ encountered from that point on.”

5.2 Implementing the keyed scheme

An engineer using this framework can proceed to design an anti-counterfeiting scheme in a slightly modified manner compared to the method presented in Section 4:

1. The engineer provides training data for damage profiles for the parts under consideration, as described in Sections 3.1-3.2.
2. Knowing which data profiles are most likely to occur, the engineer identifies the most “valuable” features for those profiles using Equations 3-5.
3. Based on these results, the engineer estimates the per-part-location performance given a set of ranked features using results data plotted as in Figure 11 and eliminates any features below an acceptable performance threshold.
4. Using the results plotted as in Figure 11 the engineer determines a range of included feature amounts that achieve low error probability for the most relevant expected damage profiles. For example, if expecting pitting damage, the engineer may enforce using 150-250 features in the final string.
5. The engineer generates a key, and then selects a subset of acceptable features of an appropriate size using this key and
generates strings as discussed above. The resulting strings are constructed from a secret permutation of a secret subset of the input features, making string prediction infeasible for an attacker with knowledge of the material system.

6 Conclusions and Future Work

In this paper, we present a method for constructing anti-counterfeiting schemes that leverages autocorrelation information inherent in micrographs of metal parts. The several hundred features that are extracted per micrograph location were analyzed for value in the face of common damage profiles. From this analysis, the engineer responsible for part traceability has access to a useful decision support tool: given requirements on data storage for each part, acceptable performance when challenging suspect parts, and the expected damage profiles for a part, an engineer can investigate which features are the most valuable (and most harmful) to include, and better understand the storage-to-confidence trade-off in designing the scheme. We have also discussed cryptographic extensions to this work for increased security in the face of determined counterfeiting adversaries.

Future work is needed to expand the library of features, damage profiles, and materials that may be considered in this framework. Since the proposed method analyzes feature value in the face of expected damage, it may be readily applied to a wider array of use cases. Further, in this study, we considered synthetic micrograph data and simulated damage conditions. Validation of the performance of this method in practical, industrial applications is needed in the future.

Acknowledgement

Parts of this work were supported by sponsors of the Center for Education and Research in Information Assurance and Security. The statements made herein are solely the responsibility of the authors.

REFERENCES


