# Side Channel Identification using Granger Time Series Clustering with Applications to Control Systems

Matthew Lee[1], Joshua Sylvester[1], Sunjoli Aggarwal[1], Aviraj Sinha[1], Michael Taylor[1], Nathan Srirama[1], Eric C. Larson[1], and Mitchell A. Thornton[1]

[1] *Darwin Deason Institute for Cyber Security, Southern Methodist University, Dallas, Texas, USA*
{*leemh, jsylvester, sunjolia, avirajs, taylorma, nsrirama, eclarson, mitch*}*@smu.edu*

Keywords: Side channel, Granger causality, Clustering, Industrial Control Systems

Abstract: Side channels are data sources that adversaries can exploit to carry out cyber security attacks. Alternatively, side channels can be used as data sources for techniques to predict the presence of an attack. Typically, the identification of side channels requires domain-specific expertise and it is likely that many side channels are present within systems that are not readily identified, even by a subject matter expert. We are motivated to develop methods that automatically recognize the presence of side channels without requiring the need to use detailed or domain-specific knowledge. Understanding cause and effect relationships is hypothesized to be a key aspect of determining appropriate side channels; however, determining such relationships is generally a problem whose solution is very challenging. We describe a time-series clustering approach for identifying side channels using the statistical model of Granger causality. Since our method is based upon the Granger causality paradigm in contrast to techniques that rely upon the identification of correlation relationships, we can identify side channels without requiring detailed subject matter expertise. A Granger-based data clustering technique is described in detail and experimental results of our prototype algorithms are provided to demonstrate the efficacy of the approach using an industrial control system model comprised of commercial components.

## 1 INTRODUCTION

In general, a *side channel* is a data source that provides unintended information leakage through a medium that was not intended to serve as a communication channel. Side channels can be characterized by signal type such as acoustical, electromagnetic, electrical, or others (Anderson, 2020). Side channel data is acquired and typically preprocessed using signal processing algorithms, statistical methods, and more recently, machine learning predictive and classification models (Ibing, 2012). Although modern data science methods are beginning to be used more in processing and exploiting side channel data, it is noted that such models, particularly machine learning approaches, have been shown to be vulnerable to cyber security attacks including those based on side channel attacks (Dubey et al., 2021). A large amount of prior research concerns the use of side channels in the field of cryptanalysis. For a history and review of such methods, see (Quisquater and Samdye, 2002). Past work includes RSA key extraction using acoustic cryptanalysis (Genkin et al., 2014), electromagnetic analysis for smart cards (Quisquater and Samdye, 2001), and many others.

One class of side channels that popularized their use in cyber security is referred to as *power analysis* wherein monitoring and processing the fluctuations in power ports of an integrated circuit or electronic system can reveal information such as the cryptographic keys employed in encryption/decryption circuits or the machine instructions being executed on a CPU core (Mangard et al., 2010; Ambrose et al., 2010; Khadem et al., 2021). A well known side channel attack utilized timing information in connection with monitoring the power signal fluctuations on a device that implements a public-key encryption algorithm (Kocher, 1996). Essentially, this method exploited the structure of software that processes a secret key's exponent one bit at time. When the exponent bit was a one, a multiply instruction is executed. Thus, by measuring the timing characteristics of the power signal and using the knowledge that multiply instructions incur more power and a longer time to execute, it was possible to extract the value of the secret key exponent value.

Indeed, most side channel methods in cyber security are time-based and we focus upon time series data in our research regarding side channel identification. A more recent example of a timing-based side channel attack is the set of exploits referred to as Meltdown (Lipp et al., 2018) and SPECTRE (Kocher et al., 2019). The general concept underlying the Meltdown and SPECTRE attacks is to use certain performance enhancing features of modern CPU cores such as the cache, branch predictor, and speculative execution circuitry to access higher privileged data. An important component of both attacks is the use of cache timing information as a side channel since differences in memory access times are present in CPU cores with modern memory hierarchies. For certain microarchitectures, an attacker can estimate the time required to reload cache data by intentionally causing certain cache operations to occur. This timing information serves as side channel information that provides an attacker with the knowledge required to determine if data is loaded into the cache by another process. This class of attacks is particularly troubling since they exploit hardware-level vulnerabilities that are not easily patched. Generally, to guard against this class of attacks, the performance-enhancing hardware must be disabled. There are a few cases where vulnerability to this class of attacks can be determined, but only for specific use-cases such as web browsers (Shou et al., 2021). Side channel attacks based upon the exploitation of microarchitectural design choices have motivated the cyber security community to begin investigating other performance enhancing subsystems such as the instruction prefetching unit (Chen et al., 2021) and it is likely that new hardware-based side channel attacks will continue to emerge.

The SPECTRE and Meltdown exploits underscore one of the central motivations for the research presented in this paper. Namely, that very detailed and domain-specific knowledge is required to identify side channels. In the case of cache timing side channels, the attacker must have detailed knowledge of the CPU core microarchitecture as well as subject matter expertise in CPU and memory subsystem architecture. When such expertise is not present, the identification of potential side channel signals is virtually impossible for all but the simplest of victim systems. Furthermore, merely possessing such subject-domain expertise is not sufficient to guarantee that all possible side channels can be identified. In most cases, cause and effect chains must be identified such that hypothetical side channel data can be theorized to possibly exist in accessible signals. When the cause and effect chains have multiple levels, that is, the chain of causal events becomes lengthy, even seasoned experts can find it challenging to determine if a set of given signals can be processed to extract useful side channel information. For these reasons, a technique to automatically determine causal relationships among a set of signals in the form of indexed or "time" series is desirable.

The field of data science and analytics has seen tremendous growth in recent years through the development of more sophisticated models for automatic prediction and classification. Common approaches are the use of Bayesian, regression, neural network, clustering, and other models. In particular, there exist a plethora of data clustering techniques; however, in a general sense, most of the data clustering methods are based upon the identification of correlation patterns or characteristics among a set of data. It is a generally accepted mantra that "correlation is not equivalent to causation." This is particularly true with respect to the identification of side channels whereby a desired event in the form of computing useful side channel information results from a lengthy chain of causal events. Furthermore, it is generally appreciated that the automatic computation, or even identification, of a true "cause and effect" relationship cannot be proven nor actually computed. However, given certain assumptions regarding the underlying theoretical model of a time series ensemble, a very restricted type of causality can be statistically inferred. This model of causal behavior was first specified by Granger in 1969 and is hence referred to as "Granger causality" (Granger, 1969). Here, we adapt the concept of Granger causality to serve as the theoretical basis of a new type of data clustering called "Granger clustering" and we apply it to the problem of identifying side channels within a set of observed time series data.

In the remainder of this paper, we summarize the concepts behind Granger causality and describe how it can be used to support our Granger clustering method for the identification of side channels. We describe the assumptions and conditions required for a candidate time series ensemble to be applicable to the approach including a description of preprocessing techniques before the clustering method is applied. We show that conventional correlation-based clustering is unsuitable for the purpose of side channel identification. To demonstrate the use of our technique, we constructed a simple industrial control system (ICS) model using commercial ICS components and we collected time series data during operation from a set of nearby motion sensors. The Granger clustering method is demonstrated to correctly identify side channels that can be used to exfiltrate data whereas the use of conventional clustering methods fail to properly identify the side channels.

# 2 SIDE CHANNELS AND ICS SECURITY

An introduction and summary of pertinent topics that support the remainder of the paper are presented here. For more details, the reader should refer to general textbooks regarding cyber security such as (Anderson, 2020) and control systems (Dorf and Bishop, 2017).

## 2.1 Side Channels

Side Channels Attacks are a well known avenue for malicious actors to identify and exploit vulnerabilities through the collection of non-functional data. This non-functional data can be utilized to infer sensitive or critical information about the system of interest. One example of such non-functional data is accelerometer and gyroscope data. When collected in environments like Android smartphones, this side channel can be exploited to infer a user's keystrokes (Cai and Chen, 2011; Javed et al., 2020). In similar experiments, a proof of concept was demonstrated with measuring perturbations in the voice during a phone call, instead of keystrokes, via smartphone accelerometer sensors (Griswold-Steiner et al., 2021). When measurements were recorded in a controlled environment, digit prediction was shown to be possible. In another case, smartphone acoustic and magnetic sensor data were collected in factory settings allowing for the identification of manufacturing devices and their processes. When collected in a 3D-printing environment, it was shown that these data can be utilized to reproduce the objects themselves (Hojjati et al., 2016). In this paper we utilize accelerometer and gyroscope data in industrial control systems (ICS). In ICS settings side channels can be used not only for communicating network data but also for detecting anomalous behavior in the ICS systems.

## 2.2 Industrial Control Systems (ICS)

Critical infrastructure, industrial and manufacturing facilities rely on computer-controlled systems that typically comprise electro-mechanical frameworks, referred to as Industrial Control Systems (ICS), to efficiently support production and processing objectives. An ICS is responsible for coordinating industrial operations so that they execute properly and on schedule. Key attributes of an ICS include safety, reliability, and in more recent years, resilience to potential cyberattacks that can disrupt functionality. In extreme cases, cyberattacks can cause damage or harm to personnel supporting the facility (Pliatsios et al., 2020; Stellios et al., 2018; Babu et al., 2017; McLaughlin et al., 2016). Detecting anomalies in an ICS can increase safety, reliability, and resilience to cyberattacks or other errant behavior. ICS rely upon numerous internal and external signals and they typically comprise a large suite of sensors, all of which can serve as potential side channel sources

Man-in-the-Middle (MITM) attacks are especially effective when carried out in an ICS environment and have the potential to create anomalous operational conditions which go undetected. ICS MITM attacks are carried out by an adversarial third party that has taken control of the communication channel between two or more endpoints in the network. Once the ICS communication channel has been successfully exploited, any messages can be modified, discarded, or replaced by an attacker. This family of attacks allow the attacker to gain complete control over an asset in the network by dropping or modifying all communication from a controller (Lan et al., 2019). Additionally, the attacker can drop or modify all communication from the asset back to the controller in a way that makes the asset appear to be operating as expected. The repercussions of such an attack could range anywhere from a small amount of downtime to catastrophic damage to equipment, infrastructure, and even personnel.

One type of ICS threat that is considered here is the injection of control packets into one or more of the networks comprising the ICS. These "packet-injection attacks" are in the form of a MITM attack that can be particularly effective for certain protocols such as ModBus since thus network protocol is implemented such that the first received control packet is executed with subsequent control packets in the transaction being ignored. Packet injection attacks are particularly troublesome since they enable an attacker to cause the ICS to perform functions desirable to the attacker while also allowing the messages sent to the Human Machine Interface (HMI) to appear to indicate normal behavior. This type of MITM attack can be carried out by inserting HMI packets that indicate normal operating conditions to accompany injected control message packets that cause undesired behavior to occur at system actuators.

The protocols, connections, and devices that enable the communication between the components in an ICS installation are supplied by various vendors and are generally inter-operable due to the use of standardized computer interfaces and networking protocols that support modern ICS implementations. In general, an ICS will demonstrate state-like behavior that characterizes its overall functionality. That is, the ICS cycles through various operating points that can

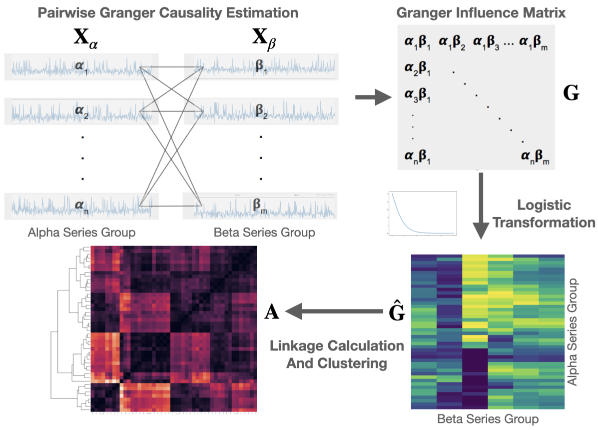


Figure 1: An overview of the proposed method of time series clustering.

be classified as a particular state. However, in many applications the state space is very large thus making it impractical to capture the complete behavior with traditional state-tracking methods.

# 3 GRANGER CAUSALITY FOR CLUSTERING

Our side channel detection method relies on a combination of Granger Causality and hierarchical clustering. Here we introduce Granger Causality and how it can be used to cluster time series.

## 3.1 Primer on Granger Estimation

The method we employ for determining measures of influence is known as Granger Causality (Granger, 1969). In general terms, Granger causality uses statistical hypothesis testing to determine if one time series is useful in forecasting another. Furthermore, Granger causality assumes that the two time series under consideration have a linear relationship with time-lagged values and additive noise present. The mathematical model for Granger causality among two time series, $x(t)$ and $y(t)$, is given in the following equation.

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \sum_{i=1}^{\tau} \begin{bmatrix} \theta_{11}(i) & \theta_{12}(i) \\ \theta_{21}(i) & \theta_{22}(i) \end{bmatrix} \begin{bmatrix} x(t-i) \\ y(t-i) \end{bmatrix} + \begin{bmatrix} \varepsilon_{11}(t) \\ \varepsilon_{21}(t) \end{bmatrix}$$

The basic idea behind the Granger model is to model the time series as linearly regressed stochastic processes. The maximum number of lagged observations determines the model order denoted by $\tau$. The model coefficients that relate the two time series are the $\theta_{jk}$ values. The model prediction errors, also known as residuals, are denoted by the $\varepsilon_{jk}$ terms. The model coefficients are random variables that are the focus of statistical inference tests to determine if they are jointly significantly different from the value zero as compared to their values when the off diagonal elements are forced to zero. We refer to the full model as unrestricted when the $\theta_{jk}$ are fully specified and the case where the off-diagonal elements are zero as the restricted model. If the prediction capability is identical for the restricted and unrestricted models, then the two time series are said to be Granger causal. Because the prediction is calculated across the entire time series, both input time series must remain stationary over time for the result to be valid.

Once the prediction parameters are estimated, an inference test is applied (typically an $F$-test of the residual variances between the two models). The application of the inference testing can vary, one approach is to use an $F$-test of the residual variances between the two models. The $F$-test compares variances of the residuals from each model, with wider variation in one model implying that the interaction variables are significant predictors. In general, the magnitude of Granger causality can be estimated by the logarithm of the corresponding $F$-statistic for this $F$-test comparison. An alternative definition called Sims Causality (Sims, 1972) can be formulated by testing the off-diagonal elements, $\theta_{jk}$, against the null hypothesis that they are zero (practically, their confidence interval includes zero). Sims and Granger causality were once thought to be equivalent, but it has been proven that Granger causality is a stronger assertion and does not necessarily imply Sims Causality. Sims causality does imply Granger causality. Practically, this distinction is subtle, and the two terms can be used interchangeably without much chance of error.

A simple extension of Granger causality can be achieved by allowing the $\theta$ coefficients to be matrices and the $\mathbf{X}(t)$ and $\mathbf{Y}(t)$ time series to be multivariate, such that each model becomes a vector autoregressive model (VAR) (Engle and Granger, 1987). The residual error across the variables is often used to understand predictive capabilities. However, the multivariate $\mathbf{X}$ and $\mathbf{Y}$ allowance comes with a computational cost. To investigate each variable, the VAR models must be created with and without each element of the vector before the $F$-statistics are calculated. The interpretation of the results is also further complicated. Alternatively, the variables can be investigated in aggregate or by groups (*e.g.*, grouped by a domain expert). The Sims causality inference test can be used in a single modeling pass because each element in the matrix $\theta_{jk}$ can be null tested for significance difference from 0. When multivariate time series include numerous variables, this test is often preferred for computational efficiency. An extension to each Causality estimate is to use an $L_1$ regularization term (Arnold et al., 2007) or mixed $L_1$ and $L_2$ regularization (Furqan and Siyal, 2016), incentivizing the values of each model toward zero. An extension of this was also proposed using forward and backward passes upon the VAR model, extending it to time series reversals (Cheng et al., 2014), though this procedure typically only has applicability in sequence estimation outside of time series.

## 3.2 Granger Clustering Methodology

To apply Granger estimation in the context of time series clustering, we use the inference estimates of Granger causality as a measure of affinity between two groups of time series referred to as the "$\alpha$ group" and the "$\beta$ group" as outlined in Figure 1. While the exact relationship between the $\alpha$ and $\beta$ time series will vary between applications, our proposed method relies on a natural dichotomy between the two groups. For datasets that do not have a natural dichotomy, Granger clustering may not be an appropriate method.

Our proposed clustering method consists of the following steps:

1. We test the stationarity of each time series using a Dickey-Fuller test (Dickey and Fuller, 1979). If necessary, higher-ordered difference versions are substituted to achieve stationarity.

2. We partition the stationary time series into two groups, $\mathbf{X}_\alpha$ and $\mathbf{X}_\beta$ that can be identified through any means. For side channel identification, a natural choice is grouping according to signal type or origin of the signals.

3. We use VAR Granger estimation to calculate the inference statistic, according to Sims causality, between each time series in $\mathbf{X}_\alpha$ and each time series in $\mathbf{X}_\beta$. This results in what we coin as the "Granger influence matrix" $\mathbf{G} = [g_{ij}]$, which is formed using the inference statistics between each pair of time series (Figure 1, upper right). Each row of $\mathbf{G}$ relates a time series in $\mathbf{X}_\alpha$ to each time series in $\mathbf{X}_\beta$, where each column corresponds to an $\mathbf{X}_\beta$ time series.

4. The Granger influence matrix is comprised of inference statistics ($p$-values) and therefore is transformed before further computation. We transform each element of the Granger influence matrix by the logistic function, where $\widehat{\mathbf{G}} = [\hat{g}_{ij}]$:

$$\hat{g}_{ij} = 1 - \frac{1}{1 + e^{-\gamma(g_{ij} - t)}}$$

Where $\gamma$ controls the steepness of the logistic mapping and $t$ controls the desired significance threshold. In our experiments we use $t = 0.05$ which indicates a statistical test with 95% confidence. In our experiments the range of $\gamma = [1, 100]$ tended to work well. Practically, this transformation is applied because inference statistics are "inverted" from their proper statistical interpretation—a small value indicates that two time series are related, but a large value indicates there might not be a strong relationship. This logistic transformation ensures that strong relationships, with $p < t$, have a $\hat{g}_{ij}$ value near unity and weaker relationships are mapped to values near zero. The $\gamma$ parameter enables users to specify confidence in the models: large $\gamma$ parameters result in a hard cutoff at $p < t$; small $\gamma$ parameters allow for a smoother transition region, incorporating weaker relationships ($p > t$) in the clustering.

5. To facilitate clustering, we calculate the pairwise cosine distance for each row of $\widehat{\mathbf{G}}$, resulting in a measure of affinity between each time series in $\mathbf{X}_\alpha$. Alternatively, we can calculate the cosine distance between each column, resulting in a measure of affinity between the time series in $\mathbf{X}_\beta$. Strictly speaking, any measure of vector similarity can be applied in this step. We choose cosine distance as an affinity measure because it is robust to the exact magnitude of two vectors (being most sensitive to the angle between vectors) and works well in our tested evaluations. The pairwise matrix of affinities is denoted as $\mathbf{A}$.

6. Finally, we use a static clustering method to cluster the time series based on $\mathbf{A}$. In our evaluations, we use hierarchical agglomerative clustering (HAC) with complete linkage (Defays, 1977). Practically, most any static clustering method may be chosen, but we find HAC provides compelling results and hierarchical relationships that are easily interpreted through a dendrogram—a method most researchers are familiar with.

## 3.3 Illustrative Example

To evaluate the performance of our Granger clustering method, we create a dataset of time series with known clustering, as shown in Algorithm 1. In this formulation, $\alpha$ group time series are generated using random distributions, and thus do not have any underlying relationship that can be used by correlation based clustering methods. We clustered these raw time series using complete-linkage HAC with a Euclidean distance measurement in order to verify this independence. For each time series, a *sinc* function

**Data:** Time series to generate, per group
**Result:** Groups of time series with known influences
1. Generate $N_\alpha$ time series from Gaussian distribution, each with 1000 points;
2. Randomly add "spikes" in each time series via *sinc* function (for structure);
3. Set $\sigma = 0.1$;
**while** *Total in* $\beta$ *group* $< N_\beta$ **do**
    Using a subset of time series from $\alpha$ group;
    $i = 0$;
    **while** $i < 4$ **do**
        Randomly shift each time series and take weighted sum;
        Add noise $\mathcal{N}(0, \sigma)$;
        Append to $\beta$ group, $i = i + 1$;
    **end**
    $\sigma = \sigma + 0.5$
**end**

**Algorithm 1:** Synthetic generation algorithm of six clusters, with increasing additive noise in each cluster.

is added at random positions to add structure to the $\alpha$ time series. Using subsets of four time series in this $\alpha$ group, $\beta$ time series are systematically generated, allowing known relationships among clusters of $\alpha$ group time series to influence time series in the $\beta$ group.

In total, 24 time series were generated in the $\alpha$ group and 18 time series in the $\beta$ group, spread among six clusters marked A-F. Notice that the $\sigma$ parameter determines how much noise is added between the $\alpha$ and $\beta$ groups for each cluster (increasing by 0.5 for each cluster). This resulted in signal-to-noise ratios (in dB) of 10, 3.5, -3.5, -7, -10 and -12.5 for clusters A-F, respectively. That is, clusters C, D, E, and F have greater noise magnitude than $\alpha$ time series. Thus, each cluster of time series becomes increasingly difficult to find amid the noise. Figure 2 (left) shows the generated time series for the $\alpha$ and $\beta$ groups and the absence of of any underlying relationships between the time series in each group. Figure 2 (right) shows the affinity matrix and clustering of the synthesized dataset. That is, HAC has been applied to reorder the affinity matrix such that the most meaningful time series cluster along the diagonal.

We see six distinct clusters in the affinity diagram and the dendrogram. Recall that six clusters were formed (A-F) with each cluster having an increased amount of additive noise. Looking at the cluster labels for the $\alpha$ time series reveals clusters of time series which all belong to the same known cluster, except one point, as marked on the diagram. This point is a noisy time series generated from cluster F (with the largest noise magnitude). Despite the $\alpha$ time series group having no discernible correlation among one another, our method is able to find the six clusters, even amidst noise with intensity greater than its own magnitude.

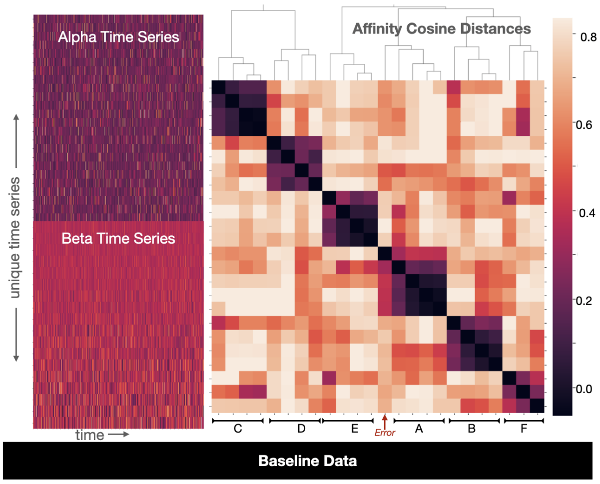


Figure 2: Evaluation of baseline dataset. Heatmaps of the time series are shown on the left and the Granger affinity matrix (after clustering) is shown on the right. Clustering is performed with $\gamma = 100$ and $t = 0.05$ for baseline data.

Having introduced and demonstrated our method for clustering time series using Granger causality, we now introduce the specific data set with which we utilized this method for side channel detection.

# 4 DATASET SELECTION AND PROCESSING

To investigate the utility of our model for identifying side channels, we leverage an existing industrial control systems dataset that uses a ModBus protocol to control the speed of a conveyor belt (Sinha et al., 2021). The dataset is ideal for investigating side channels as it contains time-series Modbus packet data along with peripheral motion sensors. These motion sensors may leak information about the state of the ICS. The network includes a Tolomatic industrial motor connected to a human-machine interface (HMI) controller with an integrated programmable logic controller (PLC), an Arduino 101 microcontroller board with integrated accelerometer/gyroscope for acceleration and gyrometric data, and an inline Raspberry Pi with two USB ethernet adapters. The Raspberry Pi acquires side channel data by handling packet collection, controlling the arduino, collecting motion data, and logging event times from data syn-

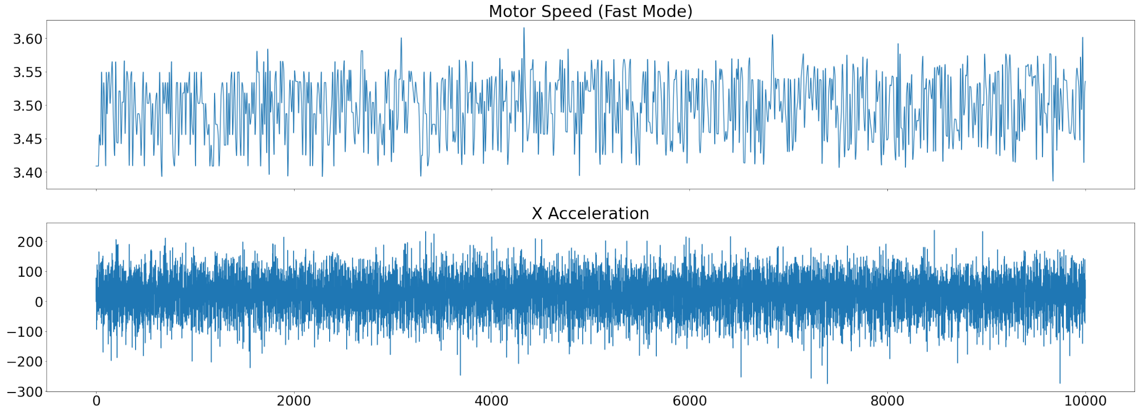

Figure 3: Examples of packet (top) and sensor (bottom) time series.

chronization. The HMI sends commands to the motor, which can turn it on or off, change or reverse its speed, etc. The HMI sends a continual stream of commands. All user input by way of the HMI is logged by the Raspberry Pi and the command is passively forwarded to the appropriate motor. The Arduino motion sensors and network traffic are continually collected by the Raspberry Pi with timestamps in order to synchronize all of the data with HMI user activity.

The sensor data is stored as Comma Separated Values (CSVs) for X, Y, and Z rotation axes for both the accelerometer and gyroscope. These axes represent the orientation of the attached sensors in the three-dimensional space; this data is collected as a constant stream as the sensor controller continuously logs data from the motor at a fixed sample rate of 10 thousand samples per second. The sensor data is logged as floating point values that represent angular velocity as degrees per second. We treat these motion time series as potential side channels that leak network information. The main data that could be leaked includes the ModBus payload. Rather than being a constant stream of data input, each payload arrives at different times to the controller. To process the data for time series clustering, the network data was preprocessed, saving each data word of the payload in the packet as separate time series. In each individual data payload there were 53 data words, leaving 53 potential time series that could have their information leaked from the side channels. However, we filtered these time series further, removing payload packets that did not change or were exactly periodic (that is, always repeating). After filtering, four data words from the 53 time series remained that encoded the speed setting of the motors (from the HMI) and the actual speed measured by the motors. Each pair of data words formed a 16-bit value for the speed setting and the measured speed—thus these four time series

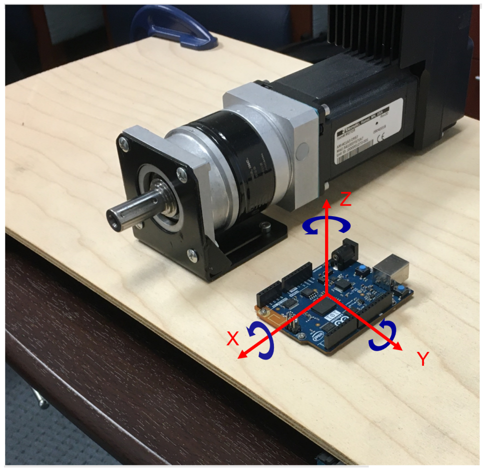

Figure 4: Orientations of sensors on Arduino relative to the motor.

could be reduced to two time series that encoded the speed into 16-bit values. We then used a Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) to upsample the packet time series to match the dimensions of the higher sample rate physical sensors. Figure 3 shows these time series examples for the motor speed packets and the acceleration data. Figure 5 shows heatmaps of all time series data, with all trials concatenated.

While the original setup also used a man-in-the-middle (MITM) device to generate network anomalies (Sinha et al., 2021), the portion of the data set used in this paper was the baseline calibration data with no anomalies. Thus we are able to use Granger time series clustering to investigate if motion sensor data can be used as a side channel to exfiltrate network information. Specifically, this exfiltrated network information was related to the encoding and measurement of motor speeds in Modbus packets.

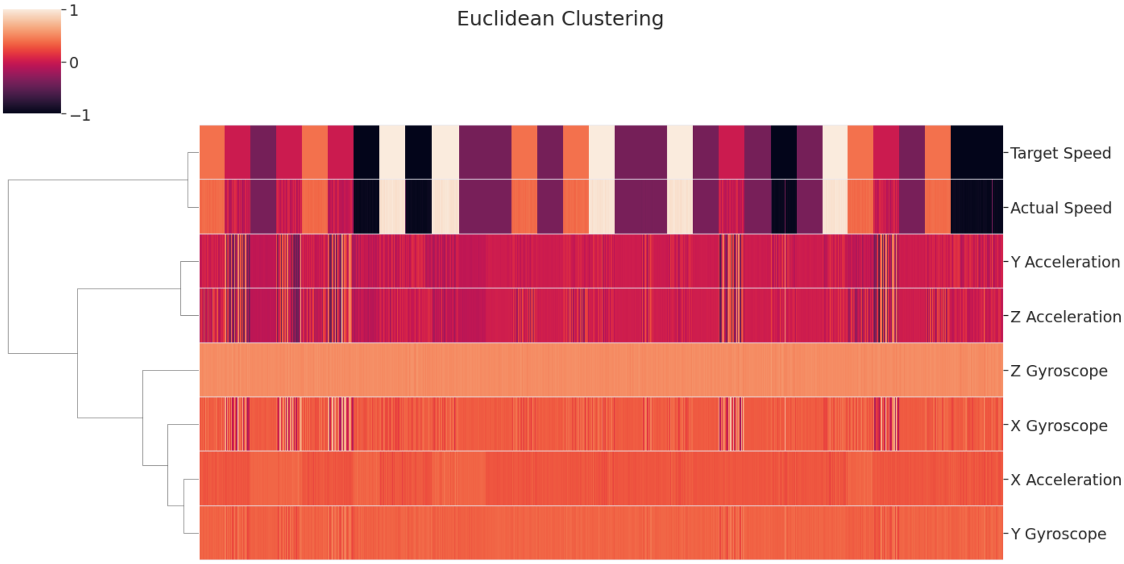

Figure 5: Euclidean hierarchical clustering of raw sensor and packet time series. Time series scaled to [-1, 1].

# 5 RESULTS

We applied our method to the data set described above from (Sinha et al., 2021) to test it's effectiveness as a side channel identifier. We also apply a traditional clustering method for comparison.

## 5.1 Traditional Time Series Clustering

We first clustered the time series using Euclidean distance as a baseline. This form of clustering only reflects the similarity of the raw values in each time series. That is, the clusters represent time series which have similar values at similar points in time. Figure 5 shows a normalized heatmap of each time series—each row is one time series and colors reflect the value of the time series. Moreover, we show a dendrogram of the Euclidean cluster linkage on the left of the plot. As we can see in Figure 5, the Target Speed and Actual Speed time series (from packet data) uniquely belong to their own cluster. This cluster is also dissimilar to the sensor time series data. This indicates that there is poor correlation between the packet data and motion sensor data, from which we might infer that these sensors do not contain any information about the control packets and thus are poor side channels. However, since feature-based clustering methods only compare time series at a single point in time, they can fail to pick up on relationships with a time lag or when the features in each time series are dissimilar. Granger causality, on the other hand, has the ability to account for these kind of time series relationships, which allows our Granger-based method to identify relationships which traditional clustering methods do not, as demonstrated in the next section.

## 5.2 Granger-Based Clustering

Our data set was made up of individual trials in which sensor and packet data were recorded while the motor was operating at different speeds. In order to create the continuous time series needed for our method, we randomly shuffled and concatenated these trials to form one continuous time series per sensor. Performing a Dickey-Fuller (Dickey and Fuller, 1979) stationarity test on all time series revealed that all time series were stationary ($p < 0.01$ for all cases). This satisfied the stationarity constraint of the Granger causality tests. These motion data time series served as our $\mathbf{X}_\alpha$ group (potential side channels). The corresponding packet data was also concatenated to form our $\mathbf{X}_\beta$ group (data from the network that could potentially be exfiltrated). The $\mathbf{X}_\alpha$ time series contained six time series (3 axes for acceleration and gyroscope) and the $\mathbf{X}_\beta$ time series contained two time series encoding the speed, as discussed in Section 4.

To prevent any possible bias in our results from ordering effects we performed our clustering on 10 independent shuffles of the data (*i.e.*, different concatenations of the time series trials) and averaged the results. With our $\mathbf{X}_\alpha$ and $\mathbf{X}_\beta$ groups created, we performed Granger clustering resulting in the Granger maps, $\widehat{\mathbf{G}}$, displayed in Figure 6 and the clusters, $\mathbf{A}$, displayed in Figure 7. Figure 6 shows the two speed related values along the horizontal axis and the motion sensor data along the vertical axis. The color of

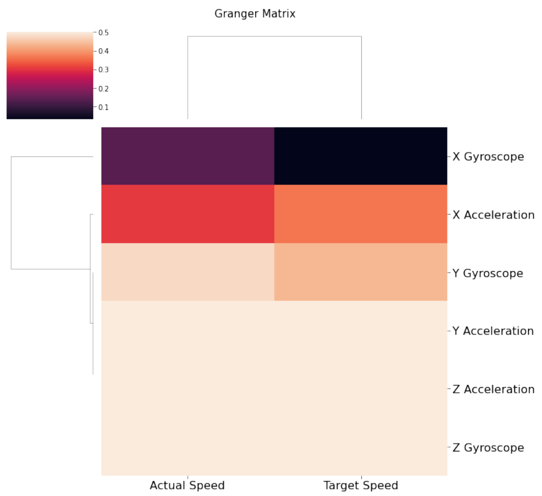


Figure 6: Clustermap of Granger matrix. Calculated as mean of resulting granger matrices from 10 independent shuffles of trials. Clustering is performed with $\gamma = 10$ and $t = 0.05$ for ICS data.

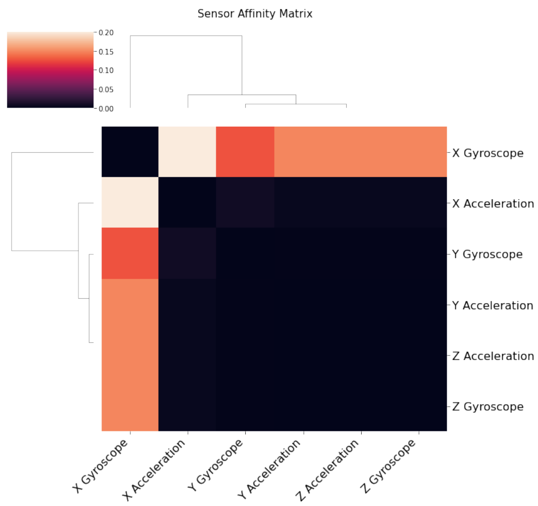


Figure 7: Granger affinity matrix. Calculated as pairwise distances between rows of Granger matrix in Figure 6.

the squares represent the strength of Granger causality found (after logistic transformation). Thus larger values indicate a stronger relationship and therefore a stronger potential for the sensor to be a side channel. Outside the plot, a dendrogram is also shown and the time series are ordered by their affinity in the dendrogram. Figure 7 shows the pairwise cosine distances that are calculated from each row of $\widehat{\mathbf{G}}$, thus revealing why the dendrogram linkages cluster. It is clear from Figures 6 and 7 that there are two distinct clusters, with the X gyroscope by itself in one cluster and the rest of the sensors in the other. As we look further down the dendrogram we see the X accelerometer also begins to separate from the other sensors, indicating that both sensors in the X direction have a weaker Granger causal relationship with the control packet data. This suggests that the X-axis gyroscope is a poor side channel for the control system, whereas significant network information is potentially leaked through physical signals in the Y and Z directions, and to some degree from the X acceleration time series.

We can corroborate this conclusion by observing the ICS system from (Sinha et al., 2021). If we look at the orientation of the sensors in Figure 4, we can see that the X direction is parallel to the axis of rotation of the motor. Thus, the speed related vibrations of the ICS would mainly occur in the Y and Z directions, corroborating that the X-axis acceleration is less powerful than Y or Z axes. Even so, Granger clustering reveals that the X-axis acceleration data can leak some information as it is strongly clustered with the other sensors. The gyroscope X direction, however, is a poor side channel. The speed of the motor cannot be predicted by this time series as it is parallel to the motion axis of the motor. This conclusion is in stark contrast to Euclidean clustering, which revealed that none of the sensor data were valid side channels.

It is important to note that the Granger clustering approach revealed these side channels without any domain expertise regarding the ICS. Whether the mechanism of data leakage was due to physical setup, magnetic coupling with the sensors, or other means, the analysis was able to reveal that packet information was leaked by a subset of sensors. Moreover, the analysis grouped the weaker response of the X-acceleration with the other sensors, revealing that the time series was potentially strong enough to be used for exfiltration. Beyond data leakage, the identified side channels could also potentially be used to define normal operation for the ICS. That is, Granger clustering reveals that a set of time series are strongly related to the ICS network operation. If a MITM attack were to occur, this relationship might be severed—for instance if the actual motor speed was spoofed, the causal sensor relationship would cease to exist. In this way, such a side channel can be used to verify normal operation of the ICS.

# 6 CONCLUSIONS

Granger clustering holds vast potential for finding side channels in ICS. In this work we showed how physical sensors can be related to control packet information using Granger-based clustering even when a correlation-based method fails to find these relationships. Physical signals such as those used in this paper as well as other non-functional information could be evaluated with this method to determine if they pose a risk as a side channel. Given the increase in complexity of ICS and the ubiquity of physical sensors in everyday devices, identifying side channels like the ones in this paper could significantly inform design and security of ICS. Future work could include the use of a non-parametric Granger causality test (Candelon and Tokpavi, 2016) and the use of non-hierarchical clustering methods for the Granger matrix.

# REFERENCES


Ambrose, J., Ignjatovic, A., and Parameswaran, S. (2010). *Power Analysis Side Channel Attacks*. ADM Verlag Dr. Müller Aktiengesellschaft & Co. KG, Saarbrücken, Germany, 1st edition.

Anderson, R. (2020). *Security Engineering, (Chap. 19 'Side Channels')*. John Wiley & Sons, Indianapolis, IN, USA, 3rd edition.

Arnold, A., Liu, Y., and Abe, N. (2007). Temporal causal modeling with graphical granger methods. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 66–75.

Babu, B., Ijyas, T., Muneer P., and Varghese, J. (2017). Security issues in scada based industrial control systems. In *2017 2nd International Conference on Anti-Cyber Crimes (ICACC)*, pages 47–51.

Cai, L. and Chen, H. (2011). TouchLogger: inferring keystrokes on touch screen from smartphone motion. In *Proceedings of the 6th USENIX conference on Hot topics in security*, HotSec'11, page 9, USA. USENIX Association.

Candelon, B. and Tokpavi, S. (2016). A nonparametric test for granger causality in distribution with application to financial contagion. *Journal of Business & Economic Statistics*, 34(2):240–253.

Chen, Y., Pei, L., and Carlson, T. E. (2021). Leaking control flow information via the hardware prefetcher.

Cheng, D., Bahadori, M. T., and Liu, Y. (2014). Fblg: a simple and effective approach for temporal dependence discovery from time series data. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 382–391.

Defays, D. (1977). An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366.

Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366a):427–431.

Dorf, R. and Bishop, R. (2017). *Modern Control Systems*. Pearson Pub., Harlow, England, UK, 13th edition.

Dubey, A., Cammarota, R., Suresh, V., and Aysu, A. (2021). Guarding machine learning hardware against physical side-channel attacks.

Engle, R. F. and Granger, C. W. (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica: Journal of the Econometric Society*, pages 251–276.

Furqan, M. S. and Siyal, M. Y. (2016). Elastic-net copula granger causality for inference of biological networks. *PLoS One*, 11(10):e0165612.

Genkin, D., Shamir, A., and Tromer, E. (2014). Rsa key extraction via low-bandwidth acoustic cryptanalysis. In *Advances in Cryptology – CRYPTO14*, pages 444 – 461. Springer Berlin Heidelberg.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438.

Griswold-Steiner, I., LeFevre, Z., and Serwadda, A. (2021). Smartphone speech privacy concerns from side-channel attacks on facial biomechanics. *Computers & Security*, 100:102110.

Hojjati, A., Adhikari, A., Struckmann, K., Chou, E., Tho Nguyen, T. N., Madan, K., Winslett, M. S., Gunter, C. A., and King, W. P. (2016). Leave Your Phone at the Door: Side Channels that Reveal Factory Floor Secrets. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 883–894, New York, NY, USA. Association for Computing Machinery.

Ibing, A. (2012). On side channel cryptanalysis and sequential decoding.

Javed, A. R., Beg, M. O., Asim, M., Baker, T., and Al-Bayatti, A. H. (2020). AlphaLogger: detecting motion-based side-channel attack using smartphone keystrokes. *J Ambient Intell Human Comput*.

Khadem, B., Ghanbari, H., and Souteh, A. M. (2021). Correlation differential power analysis attack to midori64. *CoRR*, abs/2108.11115.

Kocher, P. (1996). Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Advances in Cryptology*, pages 104–113. Springer LNCS v 1666.

Kocher, P., Horn, J., Fogh, A., , Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M., and Yarom, Y. (2019). Spectre attacks: Exploiting speculative execution. In *40th IEEE Symposium on Security and Privacy (S&P'19)*.

Lan, H., Zhu, X., Sun, J., and Li, S. (2019). Traffic data classification to detect man-in-the-middle attacks in industrial control system. *2019 6th International Conference on Dependable Systems and Their Applications (DSA)*, pages 3453–3495.

Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Fogh, A., Horn, J., Mangard, S., Kocher, P., Genkin, D., Yarom, Y., and Hamburg, M. (2018). Meltdown: Reading kernel memory from user space. In *27th USENIX Security Symposium (USENIX Security 18)*.

Mangard, S., Oswald, E., and Popp, T. (2010). *Power Analysis Attacks*. Springer Science + Business Media, LLC, New York, NY, USA, 1st edition.

McLaughlin, S., Konstantinou, C., Wang, X., Davi, L., Sadeghi, A., Maniatakos, M., and Karri, R. (2016).

The cybersecurity landscape in industrial control systems. *Proceedings of the IEEE*, 104(5):1039–1057.

Pliatsios, D., Sarigiannidis, P., Lagkas, T., and Sarigiannidis, A. G. (2020). A survey on scada systems: Secure protocols, incidents, threats and tactics. *IEEE Communications Surveys Tutorials*, 22(3):1942–1976.

Quisquater, J.-J. and Samdye, D. (2001). Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In *Smart Card Programming and Security (E-smart)*, pages 200 – 210.

Quisquater, J.-J. and Samdye, D. (2002). Side channel cryptanalysis. In *SEcurité des Communications sur Internet – SECI02*, pages 179 – 184.

Shou, C., Kadron, I. B., Su, Q., and Bultan, T. (2021). Corbfuzz: Checking browser security policies with fuzzing.

Sims, C. A. (1972). Money, income, and causality. *The American Economic Review*, 62(4):540–552.

Sinha, A., Taylor, M., Srirama, N., Manikas, T., Larson, E. C., and Thornton, M. H. (2021). Industrial control system anomaly detection using convolutional neural network consensus. *2021 IEEE Conference on Control Technology and Applications (CCTA)*.

Stellios, I., Kotzanikolaou, P., Psarakis, M., Alcaraz, C., and Lopez, J. (2018). A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services. *IEEE Communications Surveys Tutorials*, 20(4):3453–3495.