Extended Analysis, Detection and Attribution of Steganographic Embedding Methods in Network Data of Industrial Controls Systems

Tom Neubert, Eric Schueler, Henning Ullrich, Laura Buxhoidt, Claus Vielhauer
Department of Computer Science and Media
Brandenburg University of Applied Sciences
Brandenburg, Germany
surname.lastname@th-brandenburg.de

***Abstract*—Since the last decade, it is well known that Industrial Control Systems (ICS) are under attack and attackers nowadays increasingly use stealthy malware (i.e., stegomalware) implemented by steganographic embedding methods to in- and exfiltrate hidden information. Unfortunately, current mechanisms to distinguish between network steganographic embedding methods and embedded message types need improvement for a potential attribution of attackers. For the analysis of steganographic embedding methods which are utilized in stealthy malware, the work presented in this paper builds upon a state-of-the-art analysis testbed proposed earlier, which is recapitulated here. It offers the opportunity to analyze network steganographic embedding methods in ICS to elaborate methods to detect and distinguish between them to gain forensic information for attribution of potential attackers and their methods. In this work, we introduce a novel machine learning based approach to distinguish between five selected embedding methods and two embedded message types. We use the analysis testbed to evaluate and determine the accuracy of the novel approach compared to a state-of-the-art approach. In our extensive evaluation, our novel approach has shown to be able to distinguish between network steganographic embedding methods with an average accuracy of 85.7%, which is an improvement in comparison to the state-of-the-art by +5.9% and enables a more accurate attribution of attackers. Additionally, the novel approach is able to improve the accuracy of distinction between embedding method and embedded message type by +9.3% in comparison to the evaluated state-of-the-art approach.**

***Keywords*-Information Hiding; Intrusion Detection and Attribution; Network Steganography; Stealthy Malware; Industrial Control Systems**

I. Introduction

This paper is based on the conference publication in [1] and significantly extends it. Some formulations and explanations are taken directly from [1].

During the last decade, stealthy malware based on steganographic embedding techniques (i.e., information hiding techniques) is increasingly used by attackers, confirmed by recent attack vectors in [2], which show that attackers use information hiding techniques to stay undetected. Stealthy malware uses completely unobtrusive data to create hidden channels, which for example are utilized to embed malicious code or to command and control. Since the Stuxnet-Attack in 2010, it has been clear that Industrial Control Systems (ICS) are under attack with stealthy malware. In this attack, Ink-files were utilized as cover data and in-memory code injections were used to hide the attack [3]. Additionally, recent attacks like the Ukrainian [4] and the Indian power grid attack [5] demonstrate that attacks with information hiding based malware on ICS become more and more common, especially due to the motivation to stay undetected as long as possible in order to in- and exfiltrate stealthy data.

Currently, several potential information hiding attack vectors for stealthy malware with steganographic embedding techniques and potential defense mechanisms are introduced (e.g., in [6], [7], [8] and [9]).

In our earlier work [1], we presented an Analysis Testbed for Steganographic Network Data (ATSND), which enables the opportunity for comprehensive analysis and comparison of these methods to identify potential similarities, differences, and effects of the embedding methods on the cover data and to derive defense and detection mechanisms for specific embedding methods. The evaluation results of [1] show that it is possible to distinguish between analyzed embedding methods after a detection, which can lead to the opportunity to identify the context of potential attackers (attribution) with machine learning based methods.

The accuracy of the state-of-the-art approach in [1] to distinguish between embedding methods is decent, but needs improvement for a more reliable attribution. Furthermore, the approach was evaluated to distinguish between a limited number of three embedding methods, which should be extended for a more conclusive evaluation and to derive a more reliable assumption about the separation precision of an approach. Additionally, the results from [1] show that the detection of embedded types (e.g., strings consisting of invariant single characters vs. text messages consisting of heterogeneous combinations of characters) needs improvement.

Thus, the **contribution** of this paper is a significant extension of the work presented in [1] and can be summarized as follows:

- Introduction of a novel feature space to train a novel neural network driven classification model for the distinction between steganographic embedding methods and embedded message types.
- Comparison between the classification results of novel feature space and the state-of-the-art feature space from [1] to derive an assumption about a potential improvement of classification accuracy.
- Extension of evaluation by two (one novel, one from state-of-the-art) to a total of now five steganographic embedding methods and novel (extended) training and test data for more meaningful evaluation results.

In the evaluation, we analyze if there is an opportunity to distinguish between five steganographic embedding methods and if we are able to differentiate between embedded message types (invariant and heterogeneous messages) with a machine learning driven classification based on our novel handcrafted feature space in comparison to a state-of-the-art feature space.

The paper is structured as follows: In Section II, we present related work and fundamentals. In Section III, we deploy our ATSND to our specific use case. Our evaluation setup to analyze five embedding methods with ATSND, including evaluation goals, data and environment, is presented in Section IV. Section V presents the evaluation results, and Section VI concludes the paper with a summary and future work.

II. Fundamentals and Related Work

In this section, we summarize fundamentals of network steganography in ICS, describe recent steganographic attack vectors for network steganography in ICS, and present our previously introduced synthetic steganographic embedding (SSE) concept to produce synthetic steganographic network data for a fast and easy generation of network data with recent steganographic embedding methods. Furthermore, an overview of methods to analyze steganographic network data for detection and attribution purposes is given.

A. Network Steganography in ICS

"Steganography is the art and science of concealing the existence of information transfer and storage", according to [10]. Besides the various possibilities for unobtrusive embedding, such as digital media data (images, audio, video et cetera), the subdomain network steganography targets the transfer and storage of hidden information in network communication traffic. From attackers perspective, a warden (e.g., intrusion detection system) observes the network traffic and the embedding of stealthy malware should be inconspicuous in a sense that a warden would not be able to differentiate between genuine communication and communication with steganographic embedding [6]. An embedding of hidden information with steganographic techniques can be realized, for example by manipulating the network packets payload on least significant values or by modulating time intervals between specific packets [11].

Network steganography and stealthy malware in ICS are special, due to limited channel capacity and thus the lower amount of available data for potential embedding compared to traditional Information Technology (IT) networks. Furthermore, the transmitted network packets are usually smaller in ICS since only meta-data or a few values (e.g., from sensors) are transferred per packet. Additionally, ICS specific protocols like OPC UA (Open Platform Communications Unified Architecture) [12] or Modbus-TCP [13] are often encapsulated in TCP/IP (or other transport protocols), which creates the opportunity for utilizing the data fields of the ICS specific protocols in addition to TCP/IP protocol headers. It is also not uncommon for the ICS-specific payload to be transmitted unencrypted, because ICS are often considered as closed networks and not subject to attacks in practice.

Potential network steganographic embedding patterns and a related terminology are summarized in [14]. A generic taxonomy and overview with the intention of a unified understanding of terms and their applicability for network steganographic methods can be found in [10].

B. Selected Steganographic Embedding Methods for ICS

In this section, we present four relevant exemplary attack vectors with regards to their steganographic embedding methods in ICS. These Embedding Methods (EM) are selected because all of them use timestamp modulations (i.e., timing channel) to embed hidden information, which is a plausible attack vector, since every network packet includes them. We are aware that there are alternative embedding concepts like Least Significant Bit (LSB) embeddings in sensor data fields of network packets, but in the context of this article, we focus on timestamps only, because they can be applied regardless of the category of the network communication (e.g., sensor data or other) and suggest relatively higher capacity.

The state-of-the-art EM and one novel steganographic embedding method will be presented in the following subsections They will be analyzed and compared with the analysis testbed presented in Section II-D from [1].

1) Steganographic Embedding Method 1 (EM_1): The approach presented in [6] uses packet timestamps (T_i) for embedding while utilizing a dynamic encoding approach based on the hour, minute, and second values, as well as an embedding key and an initialization vector. In the approach, low-value-digits of the timestamp are manipulated. This approach is able to hide one *ASCII*-symbol in four of the five highlighted digits of a timestamp in the coding "HH:MM:SS.f**fffff**fff", where H,M,S,f stand for digits of the hour, minute, second and fractional digits of the second of the time value respectively (Example: T_i= 10:00:00.1**23456**789). The actual embedding positions are determined using the embedding key, which determines the first digit right of the floating point for the fractional second values. Converting a sequence of *ASCII*-symbols to binary values results in a bitstream BS which is embedded in chronological order into every available packet. Due to the different modulated values of the variables involved, the encoding of the output values varies in perception. The formalized algorithm description can be found in Section III-B1.

2) Steganographic Embedding Method 2 (EM_2): A quite simple and easy to comprehend embedding method is introduced in [8]. The embedding scheme assumes an attack vector with a corrupted Programmable Logic Controller (PLC) via Supply-Chain-Attack. The PLC sends delays in the microsecond range ($\mu s_1, \mu s_2, \mu s_3$) to embed a hidden message via timing delays. This means an exemplary timestamp T_i = 10:00:00.123**456**789 is manipulated on the digit positions $\mu s_1 = 4, \mu s_2 = 5, \mu s_3 = 6$. The embedding scheme converts an *ASCII*-message into a bitstream *BS*. For embedding a bit of *BS*, timestamps in three consecutive OPC UA (server) packets are altered (T_i, T_{i+1}, T_{i+2}). To stay inconspicuous, the timestamps (T_{i+3}, T_{i+4}, T_{i+5}) of the following three OPC UA packets remain completely untouched. The approach

arbitrarily chooses the digit '4' to embed bit = 0 and digit '9' to embed bit = 1. For the algorithm formalization see Section III-B2.

3) Steganographic Embedding Method 3 (EM_3): EM_3 is based on EM_2 and was introduced in [1]. EM_3 extends EM_2 with the addition of a key for a dynamic encoding and positioning of the embedding (see Section III-B3 for formalization of algorithm). EM_3 enables a more sophisticated and unobtrusive embedding, introducing dynamic cipher digits C_0 and C_1 for bit values 0 and 1, which leads to an encoding where the seed of the embedding is generated with a random number.

4) Steganographic Embedding Method 4 (EM_4): A sophisticated steganographic embedding method is introduced in [15] and was initially designed to alter transmitted sensor values in ICS. The formalization of the embedding algorithm can be found in Section III-B4. In this embedding method, each character of a message is converted into an 8 bit representation of its ASCII code c_A. Afterwards c_A is encrypted with an encryption key KE creating the encrypted character c_{AE}. Prior to embedding, four consecutive digits from a single OPC UA timestamp are transformed into a 16 bit long binary representation and the embedding takes place on the 8 least significant bits. c_{AE} is then embedded replacing the last 8 digits of the binary timestamp. The binary timestamp is transformed back into its four digit decimal representation and replaces the original (unaltered) timestamp.

5) Steganographic Embedding Method 5 (EM_5): Beyond the state-of-the-art, we present a novel steganographic embedding method EM_5 in this work. It will be described and formalized in Section III-B5.

To conclude this section, we want to align the embedding methods EM_{1-5} to the generic taxonomy for steganographic methods of [10]. EM_{1-5} can clearly be classified in the domain overlapping network and Cyber Physical System (CPS) and can be assigned to the CPS sub-taxonomy. In this sub-taxonomy, the embedding methods belong to the categories *E1.2c1. CPS Random State/Value Modulation* and *E1.3c1. CPS Least Significant Bit State/Value Modulation*.

C. Synthetic Steganographic Data Generation

Diverse and heterogeneous steganographic ICS data is needed to train and evaluate potential defense mechanisms for ICS. However, each steganographic embedding needs mostly sophisticated and complex ICS setup, which is very time consuming to assemble, and in addition, it raises various security and safety issues. Because of this, the approach of [8] introduces a concept to generate artificial steganographic network data with a limited embedding pace and a specific steganographic embedding technique based on TCP-timestamps. Based on [8], an advanced Synthetic Steganographic Embedding (SSE)-concept is presented in [7]. It offers the possibility to embed hidden information everywhere in uncompromised network packet recordings with an embedding pace near real time. This makes it possible to quickly and easily generate test data for many different embedding methods for analysis. In [8], it is assumed that the most important aspects to be simulated in network traffic are:

1) the physical network including layout and components,
2) the network traffic including types of flows, directions, protocols used, typical payloads, etc., and
3) the type and characteristics of the (steganographic) hidden channel.

Both approaches simulate only the last aspect (3) of this list, the other two are directly adopted from an uncompromised recording of a physical setup. In the presented state-of-the-art ATSND (see Section II-D), the SSE-concept from [7] is used to generate the steganographic data based on the selected steganographic embedding methods and will be described in more detail.

D. Analysis Testbed for Steganographic Network Data (ATSND)

The Analysis Testbed for Steganographic Network Data (ATSND), as originally proposed in [1], has the purpose to compare and evaluate different (network) steganographic embedding methods to offer the possibility to make a distinction between them for a potential determination or classification of attackers or embedded message types. It includes five phases:

- Phase 1 (P_1): recording of cover-data,
- Phase 2 (P_2): selection and formalization of methods,
- Phase 3 (P_3): generation of synthetic steganographic data,
- Phase 4 (P_4): selection and extraction of features and
- Phase 5 (P_5): analysis based on the features.

The phases of ATSND are recapitulated in the following subsections and visualized in Figure 1.

1) Phase 1 of ATSND (P_1): The analysis testbed begins with Phase 1 where Cover Data (CD) has to be recorded from an uncompromised laboratory ICS network setup. CD can be recorded with different hard- and software capturing tools (e.g., *Wireshark* [16]). The output file of the recording should be extracted in the *pcap* or *pcapng* file format for further processing, since these formats are well suited logging protocols for the structural recording of network data. The recording should only contain relevant traffic for a specific purpose. The cover data builds a comparative baseline of the ICS network data to illustrate the impact of the embedding by means of a comparative analysis before and after the embedding. Further, it is also the basis for the steganographic embedding with the selected embedding methods (see Phase 2) to generate the steganographic network data in Phase 3. The specific experimental setup of our laboratory ICS is described in Section IV-B.

2) Phase 2 of ATSND (P_2): Once a network cover data file is recorded, embedding methods for the analysis in Phase 5 have to be selected and should be formalized with a pseudo code representation for an uniform, comparable and comprehensible illustration. In this work, we select four embedding approaches from state-of-the-art and introduce one novel embedding method. The formalization of the embedding methods is presented in Section III.2.

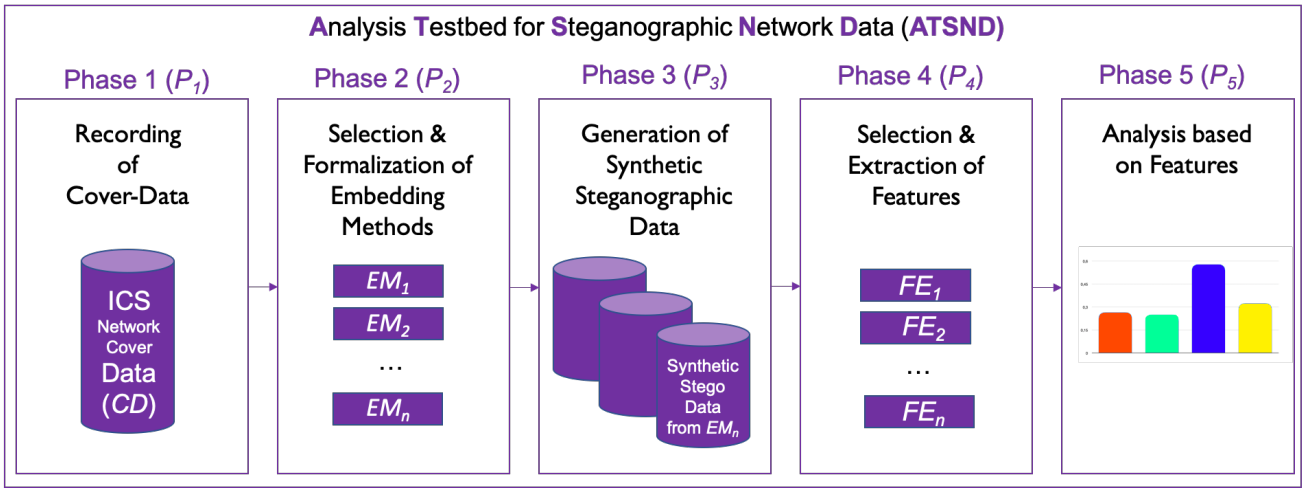

Figure 1. Analysis Testbed for Steganographic Network Data (ATSND) from [1]

3) Phase 3 of ATSND (P_3): For the creation and generation of the steganographic network data based on the embedding methods from Phase 2 (EM_1, EM_2, and EM_3), the SSE-concept [7] (introduced in Section II-C) is used. As mentioned, the SSE-concept offers the possibility to generate steganographic network data synthetically, and this results in some obvious advantages for the analysis testbed: no matter which embedding method is analyzed, it is not required to physically incorporate a corrupted, complex ICS setup in order to generate the steganographic network data containing hidden information. Thus, it is well suited because it delivers the opportunity for an easy and fast generation of steganographic network data without the need of a physical setup. The SSE-concept has the following four segments:

- Segment I: Record and Pre-Process Network Data,
- Segment II: Synthetic Embedding Option A (SEO$_A$),
- Segment III: Synthetic Embedding Option B (SEO$_B$), and
- Segment IV: Retrieval.

Segment I also deals with the recording of network data, thus Segment Element (SE) I.1 can be skipped for ATSND since the data capturing is completed after P_1. For the synthetic generation of steganographic network data, it offers two synthetic embedding options (Segment II: SEO$_A$ and Segment III: SEO$_B$). SEO$_A$ is a very fast and efficient embedding without accessing structural elements of a packet and SEO$_B$ delivers a more comfortable embedding with easier access to structural elements of a network packet based on json-objects. The retrieval in Segment IV is used to check if an embedding of a hidden message with a selected embedding method is successful. More details can be found in [7].

4) Phase 4 of ATSND (P_4): To extract features from pcap or pcapng files, the relevant structural elements of the relevant network packets should be converted into csv or txt data for processing afterwards. For this purpose, *Tshark* (*Wireshark* console application) [16] with the *-T fields -e field* option can be used to select data fields of network packets that are relevant for feature extraction and analysis. It is recommended to use handcrafted statistical feature spaces with as much discriminatory power as possible to analyze steganographic network data. This should lead to comprehensible and explainable analysis results allowing for forensic traceability.

5) Phase 5 of ATSND (P_5): Based on the extracted features from multiple embedding methods in P_4, a statistical analysis can be carried out. Therefore, various statistical computational techniques such as machine or deep learning based approaches can be taken into consideration based on the selected and extracted features. Thus, for the analysis, different data mining and machine learning tools or libraries, such as *WEKA* [17], *Orange* [18], *Tensorflow* [19] or *Keras* [20] are well suited to analyze differences and commonalities of embedding methods. Generally, the analysis can focus on different use case specific aspects, for example: detectability, attributability, embedding scheme, and more depending on the goals and objectives of a study.

E. Analysis of Steganographic ICS Network Data

A basic overview of potential methods to analyze and defend against stealthy malware based on network steganography is presented in [21]. In [1], a machine learning based approach is used to distinguish between steganographic embedding methods. The approach was initially introduced in [22] to detect network steganography in network recordings based on a handcrafted feature space with an accuracy of 92.9%. The approach performs a frequency analysis of occurrence for the digits 0 to 9 on selected positions on the packet timestamps. This feature space (FS_{SOTA}) is used for our evaluation and introduced in Section III-D.

III. Application of ATSND

As mentioned previously, we will use the Analysis Testbed for Steganographic Network Data (ATSND) from [1] (see Section II-D) for the analysis of five different embedding methods in this work. Therefore, we structure this section according to the five phases of the analysis testbed. In our specific use-case we want to evaluate if we are able to differentiate between five steganographic embedding methods and different message types with two machine learning based

classification engines for a potential attribution of attackers based on their used steganographic embedding method EM.

A. Applying Phase 1 of ATSND (Recording of CD)

As mentioned, the first phase of the ATSND concept is dedicated to the collection of network Cover Data (CD) from a laboratory ICS setup. CD can be captured with any capturing tool, as long as the output can be provided in *pcap* or *pcapng* format. Additionally, the output file should only contain relevant traffic with a specific purpose. The *pcap* and *pcapng* file formats are well suited logging protocols for the structural recording of network data. CD builds the base for the further generation of steganographic network data in Phase 3, using the selected embedding methods from Phase 2 (see Section III-B). Furthermore, CD is used as a statistical baseline of the captured ICS network data. This way the impact of each of the embedding methods can be illustrated in detail.

In order to separate training and test data, we create two separate recordings for this work. We record the training data for 25 minutes and the test data for 8 minutes in our laboratory setup which is presented in more detail in Section IV-B. In our setup, the PLC and Gateway are connected directly by an Ethernet cable, thus stand-alone packet capturing hardware [23] is used to capture the traffic between them.

B. Applying Phase 2 of ATSND (Selection and Formalization of Embedding Methods)

In this phase, it is essential to select and formalize steganographic embedding methods that shall be analyzed. The formalization helps to improve the comprehensibility of the selected embedding methods and delivers a uniform presentation of them. As previously mentioned, we select four state-of-the-art methods presented in Section II-B and one novel method (see Section III-B5). All of the algorithms work with an Array A ($A = \{T_1, ..., T_i\}$) which contains all Timestamps T_i of network packets available for manipulation in our pseudocode representation. The specific formalizations for the state-of-the-art approaches EM_1, EM_2, EM_3, EM_4 and the novel embedding method EM_5 will be described in the following subsections.

1) Formalization of Steganographic Embedding Method EM_1: EM_1 was initially introduced in [6] and takes a dynamic encoding approach while manipulating low value digits of the OPC UA timestamp. An initialization vector I and an encoding key K are used in addition to variables taken from each timestamp to encode the hidden message m with characters c. Variables D, E, F and G (meaning: see Figure 2) are all derived directly from the timestamp, as well as H ($H = \{H_0, ..., H_3\}$), which is the 4-digit field in which the encoded message characters c_E are embedded. After the encoding process is finished, the output of S decides the embedding position in H.

2) Formalization of Steganographic Embedding Method EM_2: Iterating through A, EM_2 embeds a bit of the input bitstream into 3 consecutive timestamps, encoding 0 and 1 by the digital values of 4 and 9, respectively. In the process, three different digits are used for the embedding represented in $\mu_1 - \mu_3$. Manipulated timestamps are then saved in the AM array. This is repeated for each bit in the bitstream until the end of A is reached or all bits are embedded. The algorithm was introduced in [8] and is represented in Figure 3.

Algorithm 1 Steganographic Embedding Method EM_1

$AM \leftarrow A$
$i \leftarrow 0$
$K \leftarrow 4\ Digit\ Key$
$I \leftarrow 4\ Digit\ Initialization\ Vector$
for c in m **do**
  while $i < Length(A)$ **do**
    $D \leftarrow$ Hour value of T_i
    $E \leftarrow$ Minute value of T_i
    $F \leftarrow$ Second value of T_i
    $G \leftarrow$ Value of digit 1 after floating point of T_i
    $H \leftarrow$ Value of digit 2-6 after floating point of T_i
    $S \leftarrow G \oplus DigitSum(K)\ mod\ 2$
    $O \leftarrow D \times E \times F\ mod\ 10000$
    $K' \leftarrow \sum_{n=0}^{3}((K_n \oplus (G + I_n))\ mod\ 10) \times 10^n$
    $K'' \leftarrow O \oplus K'\ mod\ 10000$
    $c_E \leftarrow c \oplus K''\ mod\ 8192$
    if $S == 0$ **then**
      $H_0, H_1, ..., H_3 \leftarrow c_E$
    else if $S == 1$ **then**
      $H_1, H_2, ..., H_4 \leftarrow c_E$
    end if
    $AM[i] \leftarrow T_i$
    i += 1
  end while
end for

Figure 2. Formalized Algorithm for EM_1.

Algorithm 2 Steganographic Embedding Method EM_2

$AM \leftarrow A$
for Bit in $Bitstream$ **do**
  for $i \leftarrow 1$ to 3 **do**
    if Bit_i is 0 **then**
      $T_i[\mu_{i\ mod\ 3}] \leftarrow 4$
    else if Bit_i is 1 **then**
      $T_i[\mu_{i\ mod\ 3}] \leftarrow 9$
    end if
    $AM[i] \leftarrow T_i$
  end for
end for

Figure 3. Formalized Algorithm for EM_2.

3) Formalization of Steganographic Embedding Method EM_3: Basically, EM_3 is an advanced and more sophisticated version of EM_2 and was introduced in [1]. It should be more challenging to detect and to attribute EM_3 in comparison to EM_2. The main difference is the key-based generation of embedding symbols (digits) C_0 and C_1, as well as the key-based variation of the embedding position j within the timestamp. The algorithm is formalized in Figure 4.

Algorithm 3 Steganographic Embedding Method EM_3

$AM \leftarrow A$
$i \leftarrow 0$
$K \leftarrow$ "$SyntheticStegoKey$"
for Bit in $Bitstream$ **do**
  for $i \leftarrow 1$ to 3 **do**
    $C_0 \leftarrow 0$
    $C_1 \leftarrow 0$
    while $C_0 == C_1$ **do**
      $C_0 \leftarrow Random(K)\ mod\ 9$
      $C_1 \leftarrow Random(K)\ mod\ 9$
    end while
    $j \leftarrow C_0 + C_1\ mod\ 3$
    if Bit_i is 0 **then**
      $T_i[\mu_j] \leftarrow C_0$
    else if Bit_i is 1 **then**
      $T_i[\mu_j] \leftarrow C_1$
    end if
    $AM[i] \leftarrow T_i$
  end for
end for

Figure 4. Formalized Algorithm for EM_3.

4) Formalization of Steganographic Embedding Method EM_4: This embedding method was introduced in [15] and its formalization is presented in Figure 5. In the formalization, the variable c represents a character of the message m and c_A the 8 bit representation of the ASCII code decimal digit of the character. It is encrypted with encryption key KE and results in an 8 bit encrypted bitstream c_{AE} of the ASCII code decimal digit, which is embedded into a 16 bit representation of a converted timestamp T_i16B (into the 8 least significant bits). After embedding, T_i16B is converted back into its initial representation.

Algorithm 4 Steganographic Embedding Method EM_4

$AM \leftarrow A$
$i \leftarrow 0$
$KE \leftarrow$ "$EncryptionKey$"
for c in m **do**
  for $j \leftarrow 0$ to 3 **do**
    $c_A \leftarrow c$
    $c_{AE} \leftarrow c_A \oplus KE$
    $T_i[Length(T_i) - j]16b \leftarrow T_i[Length(T_i) - j]$
    $T_i[Length(T_i) - j]16b \leftarrow c_{AE}$
    $T_i[Length(T_i) - j] \leftarrow T_i[Length(T_i) - j]16b$
    $AM[i] \leftarrow T_i$
  end for
end for

Figure 5. Formalized Algorithm for EM_4.

5) Formalization of Steganographic Embedding Method EM_5: Steganographic embedding method EM_5 represents a novel method. EM_5 embeds a message m into the microseconds $\mu_1 - \mu_3$ of OPC UA (server) timestamps T_i (e.g., $T_i =$ 10:00:00.123**456**789, embedding positions are marked **bold**). Before embedding each character c of m, m is saved to array *MAD* as the corresponding decimal ASCII representation of its characters c. After every element of *MAD* is embedded, first 494, then 949 are embedded into the following timestamps to signal the end of m. EM_5 was chosen for evaluation since it is a more simple algorithm which should be accurate to detect and to attribute based on the limited number of ASCII characters.

Algorithm 5 Steganographic Embedding Method EM_5

$AM \leftarrow A$
$i \leftarrow 0$
$j \leftarrow 0$
while $j < Length(MAD)$ + 2 **do**
  if $j < Length(MAD)$ **then**
    $T_i[\mu_{i\ mod\ 3}] \leftarrow MDA_j$
  else if $j == Length(MAD)$ **then**
    $T_i[\mu_{i\ mod\ 3}] \leftarrow 494$
  else if $j == Length(MAD) + 1$ **then**
    $T_i[\mu_{i\ mod\ 3}] \leftarrow 949$
  end if
  $AM[i] \leftarrow T_i$
  i += 1
  j += 1
end while

Figure 6. Formalized Algorithm for EM_5.

C. Applying Phase 3 of ATSND (Generation of Synthetic Steganographic Data)

For the synthetic generation of steganographic network data, the introduced SSE-concept is used (see Section II-C). In the evaluation, this work uses synthetic embedding option SEO_A, since it offers a much more efficient and faster embedding to generate synthetic steganographic network data based on the manipulation of hexdump elements of the network packets. All 5 selected steganographic embedding methods EM_1, EM_2, EM_3, EM_4 and EM_5 are generated with SEO_A based on the recorded cover data CD in P_1.

D. Applying Phase 4 of ATSND (Selection and Extraction of Features)

To extract features from pcap or pcapng files, the relevant structural element of the relevant network packets should be converted into csv or txt data to process it afterwards. Therefore, *Tshark* (*Wireshark* console application) [16] with the *-T fields -e field* option can be used to select data fields of network packets that are relevant for feature extraction and analysis. We recommend the usage of handcrafted statistical feature spaces with as much discriminatory power as possible to analyze steganographic network data. This should lead to comprehensible and plausible analysis results.

In this work, we use two handcrafted feature spaces to train two separate machine learning based models for our analysis in

P_5. One feature space FS_{Legacy} is used from state-of-the-art to set a baseline for our analysis goals. Additionally, we design a novel feature space FS_{Novel} to investigate if it is possible to achieve more accurate results in our analysis. The two feature spaces are presented in the following subsections. Both feature spaces analyze the last 6 digit positions of network packet timestamps because they are well suited for steganographic embedding, since every network packet has a timestamp and a potential delay in micro- and nanosecond areas is absolutely unobtrusive. A potential attack vector for our use case could look like those introduced in Section IV-B. Both feature spaces analyze multiple network packets to extract a feature vector (i.e., sample), because obviously a single packet with steganographic embedding should look unobtrusive (if not, it would not be steganographic). A measurable or quantifiable anomaly caused by a steganographic embedding can only occur by analyzing multiple network packets. In this work, we use 100 network packets to extract a sample (i.e., feature vector with label) for the feature spaces. This length (100 packets) has been selected based on state-of-the-art ([15], [22]). The optimal length with maximum separation precision can only be determined by an explorative analysis of different lengths, which is out of the scope for this work.

1) Feature Space FS_{SOTA}: The state-of-the-art feature space was introduced in [22], which performs a frequency analysis for the digits 0 to 9 on the mentioned six last and least significant digits in network packet timestamps. Thus, 10 features (values) for each analyzed digit position between 0.0 and 1.0 representing the percentage of occurrence for each digit 0 to 9 are extracted from a sample with multiple packets (as mentioned, 100 packets used to extract a sample or i.e., feature vector). The frequency analysis results in a 60-dimensional feature space which is used to train two 'legacy' multilayer perceptrons (MLP) to potentially distinguish between embedding methods and cover data (MLP_{6LG}, legacy MLP with 6 classes, based on FS_{SOTA}) and to distinguish between the embedded message types and embedding methods (MLP_{11LG}, legacy MLP with 11 classes based on FS_{SOTA}). The selected features shall be extracted for multiple samples from all embedding methods with different message types and cover data to build MLP_{6LG} and MLP_{11LG} for analysis in P_5.

2) Feature Space FS_{Novel}: Our novel feature space FS_{Novel} extends FS_{SOTA}. We add additional features based on potential artifacts caused by the embeddings. This includes the standard deviation of the digit frequencies for every digit position in the millisecond and microsecond ranges. Additionally, we calculate the standard deviation across the digit standard deviations to analyze the manipulation of single digit positions. The standard deviation over only the microseconds is also used, as embedding methods EM_2 and EM_3 only use these positions for the embedding process. In addition, the standard error of the mean of the digit distribution is calculated for each position. As a further feature, the digit transition rate is used. This feature describes the percentage of packets in which the digit at a given position changes from the preceding packet. An embedding method with a high embedding density such as EM_5 might cause digits to change less frequently. Furthermore, EM_5 changes the first digit position to a low digit. Therefore, we use the average digit value for each position. Moreover, we use Pearson's chi-squared test [24] for the distribution of digits for each position. This test describes the likelihood that an observed distribution is the result of a random sample expecting a given distribution. For the milli-, micro- and nanosecond digits of a timestamp, we expect a uniform distribution. A steganographic embedding like EM_2 uses constant values which change this uniform distribution. Additionally, the skewness of the digit distribution is calculated for every position. This describes whether the distribution is weighted towards the higher or lower end of the digits. Finally, we use the kurtosis for the digit positions, which describes the steepness in a distribution. In total, this results in a 104-dimensional feature space to train two 'new' multilayer perceptrons to potentially distinguish between embedding methods and cover data (MLP_{6NE}, new MLP with 6 classes based on FS_{Novel}) and to distinguish between the embedded message types and embedding methods (MLP_{11NE}, new MLP with 11 classes based on FS_{Novel}).

E. Analysis (P_5)

For our analysis, we will investigate if it is possible to distinguish between the five selected steganographic embedding methods (EM_{1-5}) and cover data (CD) after a potential detection of an anomaly, to potentially attribute an attacker with MLP_{6NE} and MLP_{6LG} (6-class classification challenge). Additionally, we analyze if it is possible to distinguish between embedded message types and steganograpic embedding methods with MLP_{11NE} and MLP_{11LG} (11 class classification challenge). The specific evaluation goals are presented in Section IV-A.

IV. Evaluation Setup

A. Evaluation Goals

The evaluation extends the evaluation of [1] significantly and addresses the following goals:

- G_1: Determination of the classification accuracy for MLP_{6NE} (new MLP based on novel feature space FS_{Novel}) and MLP_{6LG} ('legacy' MLP based on state-of-the-art feature space FS_{SOTA}) to analyze if and how accurate they are able to distinguish between the five selected steganographic embedding methods (EM_{1-5}) and the cover data (CD), and to investigate if new MLP_{6NE} can outscore the state-of-the-art MLP_{6LG} in this 6-class-classification-challenge.
- G_2: Determination of the classification accuracy for MLP_{11NE} and MLP_{11LG} to analyze if and how accurate they are able to distinguish between the five selected steganographic embedding methods, the two embedded message types (invariant IV message type, which means a repeated letter and heterogeneous HE message type, which means a random text message, see Section IV-C) and the cover data, and to investigate if new MLP_{11NE} can outscore the state-of-the-art MLP_{11LG} in this 11-class-classification-challenge.

The classification accuracy ACC can be determined with $ACC = (\frac{CCS}{AS}) * 100$, where CCS is the number of correctly

TABLE I
NETWORK DATA SETS FOR FEATURE EXTRACTION; STEGANOGRAPHIC DATA IS EMBEDDED SYNTHETICALLY IN REC_{CD}.

Name	Type of Recording	Embedding Method	Message Type	Hidden Message	No. of relevant Packets	No. of extracted Samples
$REC_{Train-CD}$	Cover Training-Data	-	-	-	25,613	514
$REC_{Train-EM1IV}$	Steganographic Training-Data	EM_1	invariant	'a' (repeated)	25,613	514
$REC_{Train-EM1HE}$		EM_1	heterogeneous	'IARIA-Journal-2025 ' + Lorem ipsum ... (until full)	25,613	514
$REC_{Train-EM2IV}$		EM_2	invariant	'a' (repeated)	25,613	514
$REC_{Train-EM2HE}$		EM_2	heterogeneous	'IARIA-Journal-2025 ' + Lorem ipsum ... (until full)	25,613	514
$RE_{Train-EM3IV}$		EM_3	invariant	'a' (repeated)	25,613	514
$REC_{Train-EM3HE}$		EM_3	heterogeneous	'IARIA-Journal-2025 ' + Lorem ipsum ... (until full)	25,613	514
$REC_{Train-EM4IV}$		EM_4	invariant	'a' (repeated)	25,613	514
$REC_{Train-EM4HE}$		EM_4	heterogeneous	'IARIA-Journal-2025 ' + Lorem ipsum ... (until full)	25,613	514
$REC_{Train-EM5IV}$		EM_5	invariant	'a' (repeated)	25,613	514
$REC_{Train-EM5HE}$		EM_5	heterogeneous	'IARIA-Journal-2025 ' + Lorem ipsum ... (until full)	25,613	514
$REC_{Eval-CD}$	Cover Test-Data	-	-	-	8,703	177
$REC_{Eval-EM1IV}$	Steganographic Test-Data	EM_1	invariant	'a' (repeated)	8,703	177
$REC_{Eval-EM1HE}$		EM_1	heterogeneous	'IARIA-Journal-2025 ' + Lorem ipsum ... (until full)	8,703	177
$REC_{Eval-EM2IV}$		EM_2	invariant	'a' (repeated)	8,703	177
$REC_{Eval-EM2HE}$		EM_2	heterogeneous	'IARIA-Journal-2025 ' + Lorem ipsum ... (until full)	8,703	177
$REC_{Eval-EM3IV}$		EM_3	invariant	'a' (repeated)	8,703	177
$REC_{Eval-EM3HE}$		EM_3	heterogeneous	'IARIA-Journal-2025 ' + Lorem ipsum ... (until full)	8,703	177
$REC_{Eval-EM4IV}$		EM_4	invariant	'a' (repeated)	8,703	177
$REC_{Eval-EM4HE}$		EM_4	heterogeneous	'IARIA-Journal-2025 ' + Lorem ipsum ... (until full)	8,703	177
$REC_{Eval-EM5IV}$		EM_5	invariant	'a' (repeated)	8,703	177
$REC_{Eval-EM5HE}$		EM_5	heterogeneous	'IARIA-Journal-2025 ' + Lorem ipsum ... (until full)	8,703	177

classified samples and AS is the number of all samples in the corresponding class. The results for G_1 and G_2 are presented in Section V.

B. Attack Vector and Laboratory ICS Setup of Evaluation

The recording of the cover-data in phase 1 of ATSND is done on a Fischertechnik® Lernfabrik 4.0 24V [25] model. The modeled production line consists of 2 transportation cranes, a storage rack, an environmental sensor and multiple conveyor belts, actuators and other sensors. A Siemens S7-1500 PLC controls the actuators and sensors, and connects to another network via a gateway for remote supervision. The gateway communicates directly with the Siemens-PLC using the ICS specific OPC UA protocol. Since the gateway acts as a middleman for the remote interface, its main responsibility is to collect the data of all sensors and actuators in real time. To do this, the gateway periodically requests the values of the sensors directly from the PLC. In contrast, the real time data (current and target position) of sensors and actuators is published by the PLC in shorter intervals, but only while they are active. The setup performs a close-to-reality production process including real communication involved between all components and makes use of industrial standard controllers, thus it can be considered to produce realistic and plausible ICS network traffic.

Since the OPC UA communication between the PLC and Gateway is numerous, predictable, and outward-facing (meaning leaving the Operational Technology (OT) ICS network towards Information Technology (IT) focused domains of an infrastructure), it forms a suitable cover to exfiltrate data. The fact that the communication occurs between two separate network zones would be especially beneficial for a possible attacker. A possible goal for this exfiltration could for example be the theft of confidential process information. In this attack scenario, the attacker has to manipulate the OPC UA responses coming from the PLC. This could be achieved by corrupting the control logic on the PLC itself using a supply-chain-attack.

C. Evaluation Data Sets

The cover data recorded in Phase 1 of ATSND (see Section III-A) is the base for further generation of synthetic steganographic data. In order to prevent overfitting and evaluate the MLP externally with data it has not seen before, there are two cover data sets. The larger one (REC_{Train}) consists of 25613 relevant packets and is used for training of the MLPs. For the evaluation a smaller, disjoint data set (REC_{Eval}) consisting of 8703 relevant packets is used. Our data sets are created with the SSE-concept [7], which allows a message to be synthetically embedded into a *pcap* or *pcapng* capture file. All of the embedding methods used in this paper, are modifying the recorded cover data sets synthetically. All used embedding methods modify the last digits of the OPC UA Timestamp in a network packet as described in Section III-B. In a real world attack scenario this manipulation could be achieved by a corrupted server (e.g., PLC, via supply-chain-attack) which sends timing-delays to embed the hidden information. The steganographic taxonomy introduced in [10] would categorize the used embedding methods under the *LSB state/value modulation* category.

Since one of the goals of this paper is to see if it is possible to distinguish between invariant and heterogeneous messages, we need to define the two messages to embed. The embedded invariant message consists of the repeated letter *'a'*. In order to represent the (character) similarity of natural text in the heterogeneous message, we chose to use the phrase *IARIA-Journal-2025*, followed by as much *Lorem Ispum* text as possible for each recording and embedding method. Table I shows a summary of all combinations of recording, embedding methods and embedded messages. For example $REC_{Eval-EM3IV}$ describes the recording based on the **Eval**uation cover, with the **InV**ariant message embedded by embedding method EM_3. In the following steps, the resulting steganographic data is used to extract samples of feature vectors. These are in turn used to train and evaluate our resulting MLPs.

The data used to train our MLPs for evaluation is based on the training data set from I. For the cover data training recording, we extract 514 samples (i.e., extracted feature

vectors with label). For the generated data for Goal G_1, we use the combined feature vectors of both message types for every embedding method. This means for every embedding method we have 1028 samples. For Goal G_2 we use the same cover training data, while the generated data is based only on the corresponding recording for every combination of embedding method and message type. This results in 514 samples for every training data subset. The training data setup is shown in Table II.

TABLE II
TRAINING DATA SETS USED FOR TRAINING MLP_{6LG}, MLP_{6NE}, MLP_{11LG} AND MLP_{11NE} FOR EVALUATION OF G_1 AND G_2.

Data Sets used to train MLP_{6LG} and MLP_{6NE}:				
Training Data Set Name	**Label of Samples**	**Features for MLPs extracted from:**	**Number of Samples**	**Goal**
TS_{CD}	CD	$REC_{Train-CD}$	514	G_1
TS_{EM1}	EM1	$REC_{Train-EM1IV} \cdot REC_{Train-EM1HE}$	1028 (2x514)	
TS_{EM2}	EM2	$REC_{Train-EM2IV} \cdot REC_{Train-EM2HE}$	1028 (2x514)	
TS_{EM3}	EM3	$REC_{Train-EM3IV} \cdot REC_{Train-EM3HE}$	1028 (2x514)	
TS_{EM4}	EM4	$REC_{Train-EM4IV} \cdot REC_{Train-EM4HE}$	1028 (2x514)	
TS_{EM5}	EM5	$REC_{Train-EM5IV} \cdot REC_{Train-EM5HE}$	1028 (2x514)	
Data Sets used to train MLP_{11LG} and MLP_{11NE}:				
TS_{CD}	CD	$REC_{Train-CD}$	514	G_2
TS_{EM1IV}	EM1-IV	$REC_{Train-EM1IV}$	514	
TS_{EM1HE}	EM1-HE	$REC_{Train-EM1HE}$	514	
TS_{EM2IV}	EM2-IV	$REC_{Train-EM2IV}$	514	
TS_{EM2HE}	EM2-HE	$REC_{Train-EM2HE}$	514	
TS_{EM3IV}	EM3-IV	$REC_{Train-EM3IV}$	514	
TS_{EM3HE}	EM3-HE	$REC_{Train-EM3HE}$	514	
TS_{EM4IV}	EM4-IV	$REC_{Train-EM4IV}$	514	
TS_{EM4HE}	EM4-HE	$REC_{Train-EM4HE}$	514	
TS_{EM5IV}	EM5-IV	$REC_{Train-EM5IV}$	514	
TS_{EM5HE}	EM5-HE	$REC_{Train-EM5HE}$	514	

For our evaluation of the model, we use the evaluation data set from Table I. The cover data set contains only the original recordings, resulting in 177 samples. For Goal G_1 we use the combined recordings from both message types. Each recording then contains 354 samples per embedding method. The model for Goal G_2 uses the recording for every embedding method and message type separately, so every data subset contains 177 samples. The evaluation data sets can be seen in Table III.

TABLE III
TEST DATA SETS USED FOR EVALUATION TO ACHIEVE G_1 AND G_2.

Data Sets used to evaluate MLP_{6LG} and MLP_{6NE}:				
Test Data Set Name	**Label of Samples**	**Features extracted from:**	**Number of Samples**	**Goal**
DS_{CD}	CD	$REC_{Eval-CD}$	177	G_1
DS_{EM1}	EM1	$REC_{Eval-EM1IV} \cdot REC_{Eval-EM1HE}$	354 (2x177)	
DS_{EM2}	EM2	$REC_{Eval-EM2IV} \cdot REC_{Eval-EM2HE}$	354 (2x177)	
DS_{EM3}	EM3	$REC_{Eval-EM3IV} \cdot REC_{Eval-EM3HE}$	354 (2x177)	
DS_{EM4}	EM4	$REC_{Eval-EM4IV} \cdot REC_{Eval-EM4HE}$	354 (2x177)	
DS_{EM5}	EM5	$REC_{Eval-EM5IV} \cdot REC_{Eval-EM5HE}$	354 (2x177)	
Data Sets used to evaluate MLP_{11LG} and MLP_{11NE}:				
DS_{CD}	CD	$REC_{Eval-CD}$	177	G_2
DS_{EM1IV}	EM1-IV	$REC_{Eval-EM1IV}$	177	
DS_{EM1HE}	EM1-HE	$REC_{Eval-EM1HE}$	177	
DS_{EM2IV}	EM2-IV	$REC_{Eval-EM2IV}$	177	
DS_{EM2HE}	EM2-HE	$REC_{Eval-EM2HE}$	177	
DS_{EM3IV}	EM3-IV	$REC_{Eval-EM3IV}$	177	
DS_{EM3HE}	EM3-HE	$REC_{Eval-EM3HE}$	177	
DS_{EM4IV}	EM4-IV	$REC_{Eval-EM4IV}$	177	
DS_{EM4HE}	EM4-HE	$REC_{Eval-EM4HE}$	177	
DS_{EM5IV}	EM5-IV	$REC_{Eval-EM5IV}$	177	
DS_{EM5HE}	EM5-HE	$REC_{Eval-EM5HE}$	177	

V. EVALUATION RESULTS

In this section, the determined classification results on the introduced evaluation setup for evaluation goal G_1 with MLP_{6LP} and MLP_{6NE} and for G_2 with MLP_{11LP} and MLP_{11NE} are presented.

A. Results for G_1

In G_1 we determine the classification results for the 'legacy' MLP_{6LP} based on state-of-the-art feature space FS_{SOTA} and the 'new' MLP_{6NE} based on novel feature space FS_{Novel}. This determination should show whether the presented machine learning based models are able to distinguish between the 5 presented steganographic embedding methods (EM_{1-5}) and cover data (CD). Additionally, we want to find out if the novel model can outperform the state-of-the-art approach.

TABLE IV
CONFUSION MATRIX OF CLASSIFICATION RESULTS ON TEST-DATA OF MLP_{6LG} AND MLP_{6NE} FOR G_1 (**BOLD**: CORRECTLY CLASSIFIED SAMPLES, CD = 177 SAMPLES, EM_n= 354 SAMPLES)

classified → / Actual	CD	EM1	EM2	EM3	EM4	EM5	*ACC (rounded)*
CD	**90** | **103**	9 | 0	0 | 1	43 | 45	35 | 28	0 | 0	*51* | *58*
EM1	5 | 1	**318** | **348**	0 | 0	12 | 5	19 | 0	0 | 0	*90* | *98*
EM2	1 | 1	0 | 0	**352** | **353**	0 | 0	1 | 0	0 | 0	*99* | *99*
EM3	77 | 78	23 | 0	0 | 0	**179** | **238**	75 | 38	0 | 0	*51* | *67*
EM4	24 | 40	15 | 0	0 | 0	31 | 41	**283** | **273**	1 | 0	*80* | *77*
EM5	0 | 0	0 | 0	0 | 0	0 | 0	1 | 1	**353** | **353**	*99* | *99*
						Overall Samples:	*81* | *86*

The classification results for both MLPs are presented in Table IV. We can state that both models are basically able to distinguish correctly for a majority of test samples for all classes (classification accuracies are visualized in Figure 7).

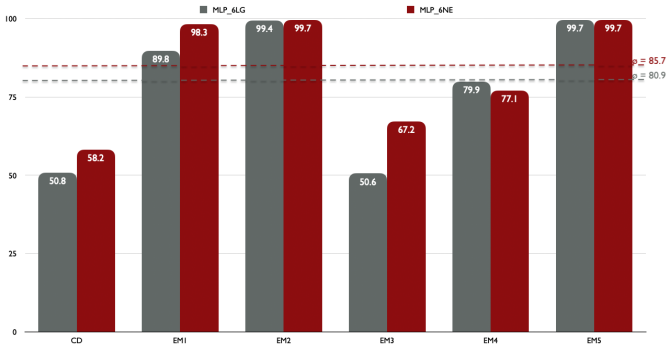

Figure 7. Classification Accuracy for MLP_{6LG} and MLP_{6NE} for each embedding method and cover data.

MLP_{6LP} reaches an overall accuracy (correctly classified samples in relation to all samples) $ACC = 80.9\%$ and MLP_{6NE} is significantly more accurate with $ACC = 85.7\%$. The classification accuracy can be especially improved with MLP_{6NE} for steganographic embedding method EM_1 ($ACC_{MLP_{6LG}} = 89.9\%, ACC_{MLP_{6NE}} = 98.3\%$) and EM_3 ($ACC_{MLP_{6LG}} = 50.6\%, ACC_{MLP_{6NE}} = 67.2\%$), and the cover data ($ACC_{MLP_{6LG}} = 50.8\%, ACC_{MLP_{6NE}} = 58.2\%$). For EM_4 MLP_{6LP} is slightly more precise in terms of classification accuracy ($ACC_{MLP_{6LG}} = 79.9\%, ACC_{MLP_{6NE}} = 77.1\%$). Both models have the same accuracy of $ACC = 99.7\%$ for EM_2 and EM_5, these methods are, as assumed,

TABLE V
CONFUSION MATRIX OF CLASSIFICATION RESULTS ON TEST-DATA OF MLP_{11LG} AND MLP_{11NE} FOR G_2 (**BOLD**: CORRECTLY CLASSIFIED SAMPLES, 177 SAMPLES PER CLASS)

classified as → / Actual	CD	EM1-IV	EM1-HE	EM2-IV	EM2-HE	EM3-IV	EM3-HE	EM4-IV	EM4-HE	EM5-IV	EM5-HE	*ACC (rounded)*
CD	**70** | **78**	5 | 1	11 | 0	0 | 0	0 | 0	8 | 28	51 | 31	4 | 10	28 | 29	0 | 0	0 | 0	*40* | *44*
EM1-IV	8 | 1	**120** | **176**	33 | 0	0 | 0	0 | 0	4 | 0	5 | 0	6 | 0	1 | 0	0 | 0	0 | 0	*68* | *99*
EM1-HE	4 | 0	22 | 0	**134** | **176**	0 | 0	0 | 0	8 | 0	6 | 0	1 | 1	2 | 0	0 | 0	0 | 0	*76* | *99*
EM2-IV	0 | 0	1 | 0	0 | 0	**172** | **174**	4 | 2	0 | 0	0 | 0	0 | 1	0 | 0	0 | 0	0 | 0	*97* | *98*
EM2-HE	3 | 3	1 | 0	0 | 0	1 | 1	**170** | **171**	0 | 0	0 | 0	0 | 2	2 | 0	0 | 0	0 | 0	*96* | *97*
EM3-IV	33 | 31	4 | 0	10 | 6	0 | 0	0 | 0	**16** | **47**	90 | 66	6 | 3	18 | 24	0 | 0	0 | 0	*9* | *26*
EM3-HE	31 | 27	5 | 1	3 | 7	0 | 0	0 | 1	15 | 42	**93** | **75**	4 | 2	22 | 26	0 | 0	0 | 0	*53* | *42*
EM4-IV	3 | 8	3 | 0	3 | 1	0 | 0	2 | 0	3 | 5	11 | 10	**120** | **113**	32 | 40	0 | 0	0 | 0	*68* | *64*
EM4-HE	19 | 25	7 | 1	8 | 2	0 | 0	0 | 0	1 | 8	20 | 21	29 | 19	**92** | **101**	0 | 0	0 | 0	*52* | *57*
EM5-IV	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	**177** | **177**	1 | 0	*100* | *100*
EM5-HE	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	0 | 0	1 | 0	**176** | **177**	*99* | *100*
											Overall Samples:	*69* | *75*

the most easiest ones to attribute correctly. Additionally, we shall notice that the accuracy for both approaches on cover data (CD) should be improved in future work, because it would trigger false positives in a real world scenario, but if we state that an attribution takes place after a previous detection (so we can exclude cover data), then especially the novel MLP_{6NE} has a decent precision to distinguish between embeddings.

B. Results for G_2

In G_2 we determine the classification results for MLP_{11LP} based on FS_{SOTA} and MLP_{11NE} based on FS_{Novel}. This determination should show if the approaches are able to distinguish between the five selected steganographic embedding methods (EM_{1-5}), the two embedded message types (IV and HE) and the cover data (CD). Additionally, we want to find out if the novel model can outperform the state-of-the-art approach.

The results for both models are shown in Table V. We can state that both models are still able to distinguish correctly between used embedding methods for a majority of test samples. Accuracy for MLP_{11LP} and MLP_{11NE} for all classes is visualized in Figure 8. Through all samples, MLP_{11LP} delivers $ACC = 68.8\%$. MLP_{11NE} delivers $ACC = 75.2\%$ overall samples and thus clearly outperforms MLP_{11LP}. The distinction between embedded message types is comparatively accurate for EM_1, EM_2 and EM_5 for MLP_{11LP}. For EM_3 the accuracy is limited, but this is explainable, due to the key-based pseudo-random embedding code generation, which makes it hard to distinguish between embedded message types.

However, on a holistic view, we can state that a distinction between embedding method and embedded message type is possible and accurate, especially with MLP_{11NE}, which is based on our novel handcrafted feature space for embedding methods with no message encryption (EM_1, EM_2 and EM_5).

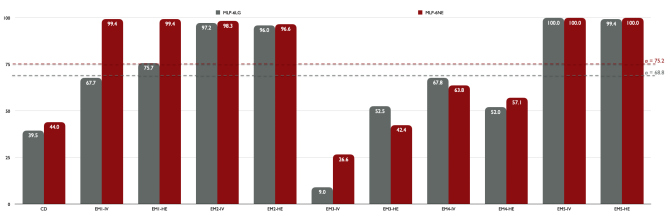

Figure 8. Classification accuracy for MLP_{11LG} and MLP_{11NE} for each message type with embedding method and cover data.

VI. CONCLUSION AND FUTURE WORK

In this paper, we analyze the possibility to distinguish between five steganographic embedding methods and two different message types based on a state-of-the-art analysis testbed for steganographic ICS network data with an extensive evaluation/analysis setup. We elaborate a novel feature space to train a machine learning driven approach with multilayer perceptron as classification engine. Our novel approach, which significantly extends a state-of-the-art-method previously presented, is able to distinguish between steganographic embedding methods with an accuracy of 85.7%, which outperforms a state-of-the-art-method by +5.9%. This creates the opportunity for a more accurate attribution, which can possibly identify the context of attackers (for example: software fingerprinting). Additionally, we are able to distinguish between steganographic embedding methods and embedded message types with an accuracy of 75.2%, which significantly improves the ability to conclude what type of message was embedded (improvement of +9.3% compared to state-of-the-art). Message type classification following a successful detection of steganographic channels may help in the attribution of different malicious payloads of stealthy malware in the future. This can be potentially achieved by differentiation between different malware code types as payload (e.g. script/shellcode vs. binary code vs. command & control instructions), deployed by different attacker groups. While steganographic communication of malware is considered to be used for illegitimate data aggregation within limited boundaries of ICS subnets, future stegomalware attacks may also make use of gateway communication, traversing borders between isolated ICS subnetworks and Information Technology (IT) network segments of the informational infrastructure of enterprises. Thus, the combination of additional forensic traces discovered on the system under attack (such as TCP/IP network traces) and steganalytic properties such as the payload type and length may allow to attribute the origin of the attack in the future for example for data in- and exfiltration via the gateway more precisely.

In future work, we would like to analyze more message types (e.g., source-code-like structures) and significantly more steganographic embedding methods. Additionally, our novel feature space has the potential to be extended for a more accurate classification. We will expand our experiments with network data from more complex ICS systems and with longer network data recordings to create a significantly larger

number of samples for training and testing. Additionally, more potential classification models based on traditional and modern machine learning techniques should be trained and analyzed to potentially improve the classification performance.

ACKNOWLEDGEMENTS

The research in this work has been performed in context of the project ATTRIBUT (https://omen.cs.uni-magdeburg.de/itiamsl/deutsch/projekte/attribut.html). This comprises in particular the conceptional design of the analysis testbed for steganographic network data and embedding method EM_3, as well as software realization in Python of all embedding methods and feature extraction. The Project ATTRIBUT is supported by funding of the Agentur für Innovation in der Cybersicherheit GmbH (Cyberagentur). The Agentur für Innovation in der Cybersicherheit GmbH did not interfere in the research process and its results. It was further supported by the SSE-concept to generate synthetic steganographic network data and embedding method EM_5 generously contributed by the project SYNTHESIS, funded by the German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV, project no. 1501666B) in the framework of the German reactor safety research program.

REFERENCES

[1] T. Neubert, B. Peuker, E. Schueler, H. Ullrich, L. Buxhoidt, and C. Vielhauer, "An analysis framework for steganographic network data in industrial control systems," *in Proceedings of SECURWARE2024 in Nice, France from November 3, 2024 to November 7, 2024; ISBN: 978-1-68558-206-7; ISSN: 2162-2116; online: https://www.thinkmind.org/library/SECURWARE/SECURWARE_2024/securware_2024_2_130_30058.html*, 2024.

[2] MITRE-ATT&CK, "Data obfuscation: Steganography," *https://attack.mitre.org/versions/v14/techniques/T1001/002/*, 2020.

[3] D. Kushner, "The real story of stuxnet," *https://spectrum.ieee.org/the-real-story-of-stuxnet, last access: 19/09/2024*, 2013.

[4] R. M. Lee, M. J. Assante, and T. Conway, "Analysis of the cyber attack on the ukrainian power grid," *SANS Institute, https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf*, 2016.

[5] I. Dragos, "Assessment of reported malware infection at nuclear facility," *https://www.dragos.com/blog/industry-news/assessment-of-reported-malware-infection-at-nuclear-facility/*, 2019.

[6] M. Hildebrandt, K. Lamshoeft, J. Dittmann, T. Neubert, and C. Vielhauer, "Information hiding in industrial control systems: An opc ua based supply chain attack and its detection," *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security, https://doi.org/10.1145/3369412.3395068*, 2020.

[7] T. Neubert, B. Peuker, L. Buxhoidt, E. Schueler, and C. Vielhauer, "Synthetic embedding of hidden information in industrial control system network protocols for evaluation of steganographic malware," *Tech. Report, arXiv, https://doi.org/10.48550/arXiv.2406.19338*, 2024.

[8] T. Neubert, C. Kraetzer, and C. Vielhauer, "Artificial steganographic network data generation concept and evaluation of de- tection approaches to secure industrial control systems against stegano- graphic attacks," *In The 16th International Conference on Availability, Relia- bility and Security (ARES 2021), August 17–20, 2021, Vienna, Austria. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3465481.3470073*, 2021.

[9] K. Lamshoeft, T. Neubert, J. Hielscher, C. Vielhauer, and J. Dittmann, "Knock, knock, log: Threat analysis, detection & mitigation of covert channels in syslog using port scans as cover," *Digital Investigation 2022 (DFRWS EU 2022), https://doi.org/10.1016/j.fsidi.2022.301335*, 2022.

[10] S. Wendzel, L. Caviglione, W. Mazurczyk, A. Mileva, J. Dittmann, C. Krätzer, K. Lamshöft, C. Vielhauer, L. Hartmann, J. Keller, T. Neubert, and S. Zillien, "A generic taxonomy for steganography methods," *ACM Comput. Surv. vol.57, no.9, https://doi.org/10.1145/3729165*, 2025.

[11] W. Mazurczyk, S. Wendzel, and K. Cabaj, "Towards deriving insights into data hiding methods using pattern-based approach," *ARES 2018, 13th International Conference on Availability, Reliability and Security; Hamburg, Germany, August 27 - August 30, ISBN: 978-1-4503-6448-5*, 2018.

[12] OPC-Foundation, "Unified architecture," *https://opcfoundation.org/about/opc-technologies/OPCUA/*, 2008.

[13] ACROMAG-Incorporated, "Introduction to modbus tcp/ip," *https://www.prosoft-technology.com/kb/assets/intro_modbustcp.pdf , last access: 19/09/24*, 2005.

[14] S. Wendzel, L. Caviglione, W. Mazurczyk, A. Mileva, J. Dittmann, C. Krätzer, K. Lamshöft, C. Vielhauer, L. Hartmann, J. Keller, and T. Neubert, "A revised taxonomy of steganography embedding patterns," *In the Proceedings of 16th International Conference on Availability, Reliability and Security (ARES 2021), Article No.: 67, Pages 1 - 12, August 17–20, 2021, Vienna, Austria. ACM, New York, NY, USA, 12 pages. https://doi.org/10. 1145/3465481.3470069*, 2021.

[15] K. Lamshoeft, C. Kraetzer, J. Dittmann, T. Neubert, and C. Vielhauer, "Information hiding in cyber physical systems: Challenges for embedding, retrieval and detection using sensor data of the swat dataset," *In Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security (IHMMSec '21), pp. 113 - 124, June 22–25, 2021, Virtual Event, Belgium. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3437880.3460413*, 2021.

[16] Wireshark-Foundation, "About wireshark," *https://www.wireshark.org/about.html*, 2024.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl. 10.1145/1656274.1656278*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[18] J. Demšar, T. Curk, A. Erjavec, Črt Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, and B. Zupan, "Orange: Data mining toolbox in Python," *Journal of Machine Learning Research, http://jmlr.org/papers/v14/demsar13a.html*, vol. 14, pp. 2349–2353, 2013.

[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," *https://www.tensorflow.org/*, 2015.

[20] F. Chollet *et al.*, "Keras," *https://keras.io*, 2015.

[21] L. Caviglione, "Trends and challenges in network covert channels countermeasures," *Applied Sciences*, vol. 11, 02 2021.

[22] T. Neubert, A. J. C. Morcillo, and C. Vielhauer, "Improving performance of machine learning based detection of network steganography in industrial control systems," *In the Proceedings of 17th International Conference on Availability, Reliability and Security (ARES 2022), Article No.: 51, pp. 1 - 8, August 23– 26, 2022, Vienna, Austria. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3538969.3544427*, 2022.

[23] Hak5, "Packet squirrel mark ii," *https://shop.hak5.org/products/packet-squirrel-mark-ii*, 2025.

[24] K. Pearson, "X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, vol. 50, no. 302, https://doi.org/10.1080/14786440009463897*, 2009.

[25] Fischertechnik, "Instruction material for the learning factory industry 4.0 24v," *https://www.fischertechnik.de/en/industry-and-universities/technical-documents/simulate/training-factory-industry-4,-d-,0-24v*, 2025.