EasyChair Preprint

№ 6884

# Non-Technical Losses Detection in Distribution Grids Using LSTM Networks

Soodanbek Kasymaliev, Victoria Erofeeva and David Pozo



October 19, 2021

# Non-Technical Losses Detection in Distribution Grids using LSTM Networks

Soodanbek Kasymaliev
*Skolkovo Institute of Science and Technology (Skoltech)*
Moscow, Russia
soodanbek.kasymaliev@skoltech.ru

Victoria Erofeeva
*Skolkovo Institute of Science and Technology (Skoltech)*
Moscow, Russia
v.erofeeva@skoltech.ru

David Pozo
*Skolkovo Institute of Science and Technology (Skoltech)*
Moscow, Russia
d.pozo@skoltech.ru

***Abstract*—Non-technical losses (NTL) constitute a major issue in many countries. NLT can be considered as a bad data detection problem. Thus, classical approaches like the weighted least square method and statistical tests can be used to detect and identify bad data regarded from NTL. Classical approaches are suitable tools when the topology of the network and its parameters are known. While this assumption is widely accepted in transmission grids, it is not the case in distribution grids, where grid reconfiguration is common, and parameters have a significant dependence on the ambient conditions.**

**In this paper, we leverage the latest advances in mathematical and computational tools to detect NTL in distribution grids. Thus, NTL detection can be implemented in an automated system that does not require human interaction. We use off-the-shelf machine learning algorithms for dealing with it. In particular, we introduce a new architecture that combines different types of deep neural networks, such as convolutional and recurrent neural networks. A thoughtful set of simulations over a realistic dataset is performed and compared with other model-free machine-learning approaches, namely, support vector machine, random forest, and gradient boosted trees.**

***Index Terms*—Non-technical Losses, Long Short Term Memory Network, Deep Neural Network, Machine Learning, Random Forest, Support Vector Machine**

## I. Introduction

The day-to-day operations of modern society strongly rely on electricity. Electric power grids enable transmission, distribution, and supply of electricity to customers from generation infrastructures such as conventional thermal power plants or photovoltaic panels. Electricity transmission and distribution includes two types of losses, namely, technical losses (TLs) and non-technical losses (NTLs) [1]. TLs occurring during transmission and distribution of energy are mainly due to the dissipation of power caused by the internal resistance of power grid components, such as overhead transmission lines, underground transmission cables, and transformers. Generally, NTLs are defined as the amount of unaccounted electricity arising from various factors. They mainly occur because of energy meter malfunctions, energy theft, errors and irregularities in billings [2]. The expenses associated with the NTLs have to be covered by electric utilities and/or legitimate consumers [3].

978-1-6654-4875-8/21/$31.00 

NTLs are among the most significant issues for electricity distribution utilities around the globe. They considerably affect the economies, reducing the profit of electricity suppliers, endangering the stable and reliable operation of electric power grids, and eventually, increasing the usage of natural resources which consequently increases pollution [4]. NTLs are a particularly major problem in developing countries where their proportion can reach up to 40% of all the electricity supplied [5]. For instance, In 2013, NTLs were up to US$46 million in Jamaica, reflecting 18% of the overall fuel bill [6].

### A. Motivation

Broad integration of the advanced metering infrastructures into electric distribution grids has enabled electric utilities to address the issue of NTLs more efficiently through the observation of the measurements obtained from the smart meters (SMs) [7]. SMs, however, are exposed to a number of vulnerabilities from the security point of view, which can consequently cause an occurrence of NTLs. This has generated a considerable interest of many researchers around the globe to take advantage of SMs capabilities to tackle the problem of NTLs.

Both academia and industry have been demonstrating an increasing interest in finding acceptable approaches to detect NTLs. Techniques for NTL detection can be categorized as *data-driven*, *network-oriented* (aka physics-aware) and *hybrid* [8]. Data-driven and network-oriented approaches imply the utilization of energy consumption data (measurements). The key difference between them is that the network-oriented approach additionally utilizes power grid data, i.e., network topology and network parameters. On the other hand, data-driven approaches only need consumer-related data, i.e., power consumption profiles, categories of consumers, etc. In that sense, it can be defined as model-free approaches. The hybrid approaches are the combination of the aforementioned two methods. Figure 1 shows the three principal categories of NTL detection methods.

Data-driven and network-oriented methods can be further broken down into subcategories according to NTL detection methods' algorithmic core idea. The former can be split into supervised and unsupervised, and the latter is divided into

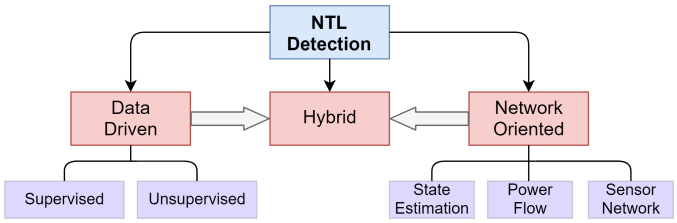


Figure 1. Categorical structure of NTL detection methods

three following subcategories: state estimation (SE), power flow, and dedicated sensors for fraud detection [8].

Machine learning (ML) algorithms are attracting the attention of many researchers for dealing with the NTL identification problem. Promising results has been obtained from recent research on ML-based algorithms, emphasizing the significance of that.

This paper is inspired by the limitation of network-oriented methods for the NLT problem and by the growing number of contributions to the state-of-the-art in new ML approaches. The synergy of the new NLT problem and ML tools has risen with the number of dedicated works. Many of those works have still implemented feature-engineering-driven approaches to extend existing ML approaches to address NTL detection problems. In this work, we focus on data-driven (i.e., model-free) approaches. In particular, we propose a deep learning network architecture in combination with convolutional recurrent neural networks.

### *B. Literature Review*

Numerous research papers have been published recently that address the problem of NTL detection. A thorough survey of NTL detection challenges is presented in [4]. Various NTL detection techniques have been analyzed and compared, such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), genetic algorithms, and other methods. Moreover, [4] shortlists several issues on the NTL detection problem. For instance, the authors discuss the need for a more detailed comparative analysis of different techniques that tackle the class imbalance domain, suggest creating a dataset for NTL detection that is publicly available, and highlight the need for adequate metrics to assess the results of NTL detection methods. The survey, however, does not include network-oriented methods.

SVM is used in [9] to identify NTL in a dataset with unevenly distributed class labels. The authors reported a significant increase of a hit rate from 3% to 60%. The article addresses NTL identification in cases when the consumption profile changes in an abrupt manner. The gradual changes in the load demand profiles were omitted. In [10], *Glauner et al.* have analyzed monthly consumption profiles of a large dataset consisting of about a million customers to detect NTL using SVM, fuzzy logic, and Boolean rules. Authors showed that SVM and optimized fuzzy models trained on datasets with relatively even class balance outperform Boolean rules. However, authors did not take into account the fact that different consumers may come from different distributions as they were located in different places; hence their consumption behavior might vary considerably. In [11], *Breno et al.* proposed an ANN on a class-labeled dataset, which contained almost 23 thousand customer records. An averaged consumption for the last 12 months was used along with other features, such as geographical location, voltage levels, and customer category. The authors reported accuracy of 87.17% along with precision and recall being 65.03% and 29.47% correspondingly. It has to be noted that in the NTL detection domain, the recall metric may be of higher importance than accuracy.

In [12], load profiles of 5 thousand Brazilian customers are studied. Each object (customer) comprises of 10 features containing charged demand, maximum demand, installed power, etc. The authors reported test accuracy scores of 96.28%, 96.20%, and 94.48% for the SVM, k-nearest neighbors, and ANN, respectively. Although the results seem optimistic, the work has been limited to just an accuracy metric. A dataset containing 3.6 million customers, with 820 thousand inspection results, e.g., inspection date, irregularity check, NTL kind, and comments, was analyzed in [13]. Four machine learning classifiers were used in the training; namely, decision trees (DT), random forest (RF), gradient boosted tree (GBT), and linear SVM. RF showed the best result among other classifiers, reaching the area-under-the-curve (AUC) score value of 0.66. In a recent study, a combination of CNN and an LSTM architecture was employed to detect NTL [14]. The authors used a class-labeled dataset of 10 thousand costumers. However, due to a highly imbalanced dataset, a sampling-based approach was utilized; a part of the data was generated using synthetic minority oversampling technique. An accuracy score of 89%, with respectively 90% and 87% for precision and recall scores, have been reported. A wide and deep CNN architecture is proposed in [15]. The authors have used a dataset that included daily electricity consumption of about 42 thousand customers over a period of approximately 3 years. Using 2D convolution layers for univariate time-series input data seems well-founded. A combination of a CNN and RF was proposed by [16] to improve the detection accuracy for the NTL. Particular emphasis was put on the overfitting problem. Recent works [17] has addressed simple LSTM architectures for the NTL detection.

Finally, we can find works considering more traditional methods in distribution grids. For instance, a weighted least squares method for the state estimation problem is suggested in [18] [19] to determine the loading of medium/low voltage transformers using the data obtained from three-phase voltage, current, and power measurements. The occurrence of NTL may be assumed if a considerable difference between measured and estimated values are observed.

### *C. Paper Contribution and Organization*

This paper's main contribution is to propose a new architecture of supervised machine learning model for NLT detection in distribution grids. In particular, we introduce a deep learning (DL) network architecture based on the combination of a

convolutional neural network and a long-short-term-memory network. The performance of the proposed DL architecture is compared with other recently published approaches for NTL problem, namely, support vector machine, random forest, and gradient boosted trees [15] [14].

The remainder of this work is organized as follows. Section II introduces the mathematical preliminaries for DL architectures. The proposed model for NLP is presented in Section II-D. Section III illustrates the proposed methodology on real-life data. Finally, the conclusion is presented in section IV.

## II. Mathematical Preliminaries for Deep Learning Architectures

There is a wide variety of opinions on deep learning algorithms, ranging from considering them a solution for any problem to as hype with expiration time. None of these extremist visions are probably right; however, it is clear that DL has undoubtedly improved performance in solving several problems, such as image classification [20], mainly due to the availability of new large data sets and the increase of computational power.

### A. Deep Neural Networks

Deep learning algorithms and deep neural networks architectures have been increasingly used for solving popular machine learning problems (e.g., classification) [21]. Deep neural networks (DNN) represent a comprehensive architecture of artificial neural networks (ANN). It is usually described by a long sequence of *layers*. DNN approximates an unknown underlying mapping function $\mathcal{F}$ from input vector $\mathbf{x}$ to output vector $y$, i.e.,

$$\mathcal{F}(\mathbf{x}, \boldsymbol{\theta}) \approx \mathbf{y}, \tag{1}$$

where $\boldsymbol{\theta}$ are parameters of function $\mathcal{F}$. The process for finding parameters $\boldsymbol{\theta}$ given a set of data composed of input-output pairs, $\{x_j, y_j\}_{j=1}^{d}$, where $d$ is the size of a dataset, is known as *learning or network training*.

DNN limits the infinite possible families of function $\mathcal{F}$ for describing data to a set of functions that can be described as layers, i.e., forming a network. It brings computational advantages not only when evaluating function $\mathcal{F}$ given some input, but also when learning parameters $\boldsymbol{\theta}$. The output vector of one layer will be the input vector of the next layer. In a general form, layer $i$ can be described as

$$\mathbf{l}_i = \gamma(\mathbf{W}_i \mathbf{l}_{i-1} + \mathbf{b}_i), \tag{2}$$

where $\mathbf{l}_0 = \mathbf{x}$ for the first layer, and $\mathbf{l}_L = \mathbf{y}$ for the last layer. The parameters $\boldsymbol{\theta}$ are represented by weight matrices $\mathbf{W}_i$ and bias vectors $\mathbf{b}_i$. The multidimensional linear function $\mathbf{W}_i \mathbf{l}_{i-1} + \mathbf{b}_i$ is simple but limited to a small family of functions. Thus, it usually is transformed by a differentiable non-linear function $\gamma$, known as *activation function*. Typical activation functions are the sigmoidal function, the $\tanh$ function, and the Rectified Linear Unit (ReLU) functions. At this point, it is worth to highlight the usefulness of activation functions to extend the capabilities of a DNN to map non-linear process, see for instance, the sigmoidal function that transform input values from $-\infty$ to $\infty$ into $[0, 1]$ values.

With a large enough number of layers (i.e., parameters) and appropriately selected activation functions, we could build a huge number of functions (input-output maps) that can match with data observations. However, there are other types of layers and structures that could provide additional benefits such as convolutional layers and recursive (feed-backward) ones.

### B. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) were particularly designed to tackle the image recognition problem, where the traditional ANNs perform poorly. The key structural difference between CNNs and ANNs is the inclusion of convolution layers. CNNs has the ability of the former to extract meaningful features from data. Normally, CNN is composed of multiple layers. They are the convolution layers and pooling layer.

The conceptual structure of a CNN is depicted in Figure 2.

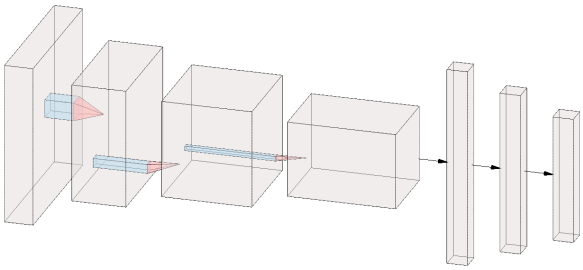

Figure 2. A conceptual CNN architecture

The main idea behind CNNs is the *convolution layers*. They perform the feature extraction by applying appropriate filters, also referred to as kernels, and moving it along the whole data vector. Mathematically, this can be represented as:

$$(f \odot u)(t) = \int_{-\infty}^{\infty} f(\tau) u(t - \tau) d\tau \tag{3}$$

where $f$ and $u$ are some functions, $\tau$ determines the increment size of the shift along $t$. The convolution between $f$ and $u$ is done by sliding along $t$ by $d\tau$.

Next, *activation functions* are applied. They are nonlinear, which essentially makes the learning process possible. Usually, a rectified linear unit (ReLU) is employed in the convolution layers. ReLU activation function is given as $f(x) = max(0, x)$.

After, a *pooling layer* is applied. Its goal is to reduce the dimensions of the convoluted layer. The most frequently used one is the max-pooling layer. In a way, it is similar to the convolution layer, however, when sliding over the convoluted layer, it captures the largest element within the sliding window (filter).

### C. Long-Short-Term-Memory Networks

LSTM networks are a particular kind of recurrent neural networks. They were devised to solve the problem of the latter which was related to numerical issues caused by the *vanishing gradients* [22]. LSTMs are able to perform the propagation of a considerable amount of information through all stages from the beginning till the end. The general structure of a single LSTM block is presented in Figure 3. An essential element of

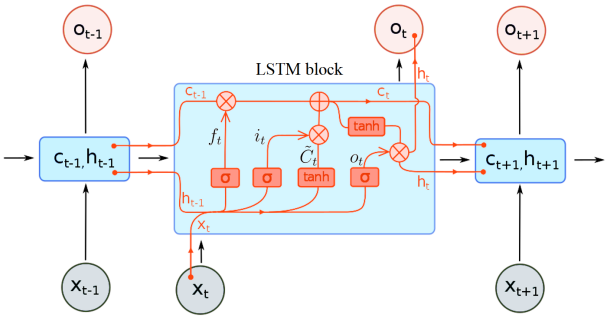


Figure 3. A general LSTM structure

LSTMs is the cell state, $\mathbf{C}_t$. It carries information throughout the entire chain which encounters some linear transformations along the way. These linear transformations are performed by structures named gates.

There are three gates in an LSTM block. The *forget gate* is responsible for deciding what portion of information should be passed further, which is composed of a sigmoid layer. It outputs a real number between 0 and 1.

$$\mathbf{f_t} = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{4}$$

If it outputs 0, no information is passed further, and if it outputs 1, then the information remains unchanged. The next gate, named the *input gate* together with $\tilde{\mathbf{C}}_t$ layer decides what information is stored in the cell state:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{5}$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \tag{6}$$

Now, the cell state is given as:

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \tag{7}$$

Finally, the *output gate* $\mathbf{o}_t$ together with the current cell state $\mathbf{C}_t$ which goes through $\tanh$ layer, form the current hidden state $\mathbf{h}_t$ that is passed to the next LSTM block:

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{8}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \tag{9}$$

### D. Proposed CNN-LSTM architecture

The model used in this work combines CNN and LSTM. Its general structure is shown in Figure 4. The input is a time series of size $b \times 8760 \times 1$, where $b$ is the batch size. The number of filters of the convolution layers is 64, 16, and 4 for the first, second, and third convolution layers respectively. The window sizes of the pooling layers are 4, 4, and 2 similar to the convolution layers. A flattening layer is essentially a fully-connected ANN, its output serves as an input for the LSTM block. The dropout layer is used for regularization purposes to prevent overfitting.

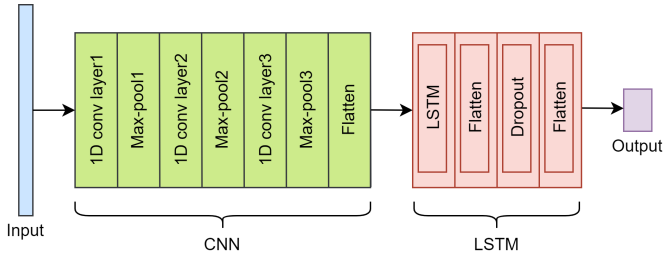


Figure 4. A general CNN-LSTM structure proposed for the NTL detection problem

## III. Case Study

### A. Data

The dataset used in this work contains an hourly consumption of electricity over a period of one year (8760 hours). The data was collected from a real distribution network in Midwest, U.S. [23], which has 1120 customers. However, for privacy reasons, the consumption of an individual consumer was not disclosed. Instead, several consumers that were connected to the same secondary distribution transformer are displayed as one. The network model has 240 nodes. However, the dataset has only 193 nonzero consumption profiles.

As was stated earlier, the main focus of this paper is DL method for the NTL detection. However, in the literature, there are several research papers that utilized standard ML methods, particularly, SVM is one of the most frequent encounters. Thus, as a benchmark, three traditional ML methods are considered, namely, SVM, RF, and GBT.

In Figure 5, a randomly selected power consumption profile is depicted.

### B. Validation Metrics

NTL detection can be considered as the detection of fraud. Usually, the data provided for such problems are highly

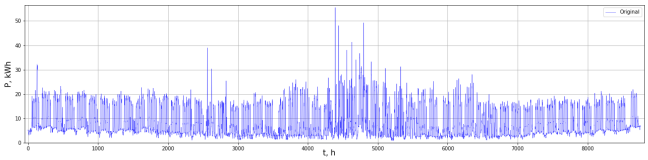

Figure 5. A power consumption profile of a benign user

imbalanced. It is possible to even have 99% to 1% ratio of respectively, benign and malicious classes [24]. The dataset used in this work is synthetically labeled, hence it is balanced. The most widely used metric is accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{10}$$

where, TP, TN, FN, and FP, respectively, are true positive, true negative, false positive, and false negative values.

We also use the following metrics:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{11}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

The above metrics are used due to the randomly distributed target labels over the dataset, which may lead to an uneven proportion of the classes in the train set and test sets.

### C. NTL Simulation

As supervised ML methods were used, the dataset needed to have target labels. Let $\mathcal{E}_t^i$ denote the initial power consumption profile of $i$-th customer at time $t$. A fraudulent behaviour of this consumer is imitated by $\mathcal{E}_t^i \cdot (1 - a_t)$. The following so called attack functions were implemented to simulate fraudulent behaviour of consumers [25]:

$$\text{BASE:} \quad a_t = \begin{cases} a^{\max}, & t \geq T^{\text{start}} \\ 0, & t < T^{\text{start}} \end{cases} \tag{13}$$

$$\text{INTERRUPT:} \; a_t = \begin{cases} a^{\max}, & t \in T^{\text{fraud}} \\ 0, & t \notin T^{\text{fraud}} \;\&\; t < T^{\text{start}} \end{cases}$$

$$\text{SMART:} \quad a_t = \begin{cases} a^{\max}, & t \geq T^{\max} \\ b(t - T^{\text{start}}), & T^{\text{start}} \leq t \leq T^{\max} \\ 0, & t < T^{\text{start}} \end{cases}$$

$$\text{COMBINED:} \; a_t = \begin{cases} a^{\max}, & t \geq T^{\max} \;\&\; t \in T^{\text{fraud}} \\ b(t - T^{\text{start}}), & T^{\text{start}} \leq t < T^{\max} \;\& \\ & \quad \& \; t \in T^{\text{fraud}} \\ 0, & t < T^{\text{start}} \;\&\; t \notin T^{\text{fraud}} \end{cases}$$

$T^{\max}$ corresponds to the time when $a^{\max}$ occurs. The latter corresponds to the largest $a_t$. $T^{\text{fraud}}$ is a set of randomly selected days from $[T^{\text{start}}, T^{\text{start}} + 1, \ldots, 365]$. The definitions for the rest of the terms in (13) are summarized in Table I which is taken from [25].

Each type of attack function applied to the power consumption profile is shown in Fig. 6

### D. Training and Validation Results

The given dataset has only 193 examples. Using the attack functions described above, we generate fraudulent users labeling roughly half of the given dataset as NTL case. We apply each attack function separately as our dataset is relatively small and use 80% of the dataset for training, and 20% for testing the algorithms. First, we fit the raw data, e.g., no feature extraction is implemented. The results are presented in Table II. In addition, Table II summarizes results from the benchmark ML methods, SVM, RF, and GBT. For clarity, we have highlighted in gray the best performer for each validation metric.

We observe that among the traditional models, SVM stood out the most, performing even slightly better than CNN-LSTM on the BASE attack function while it is not the case for the rest of the attack functions.

TABLE I
NTL MODEL PARAMETERS [25]

| Parameter | Definition |
|---|---|
| Fraud starts $T^{\text{start}}$ | The day of year (1-365) a consumer starts committing fraud. Days before $T^{\text{start}}$ are free of fraud. |
| Attack Intensity ($a_t$) | The percentage of energy stolen over actual energy consumed at time $t$. |
| Ramp slope ($b$) | The slope (%/day) of the "Smart" attack indicates how fast attack intensity increases. |

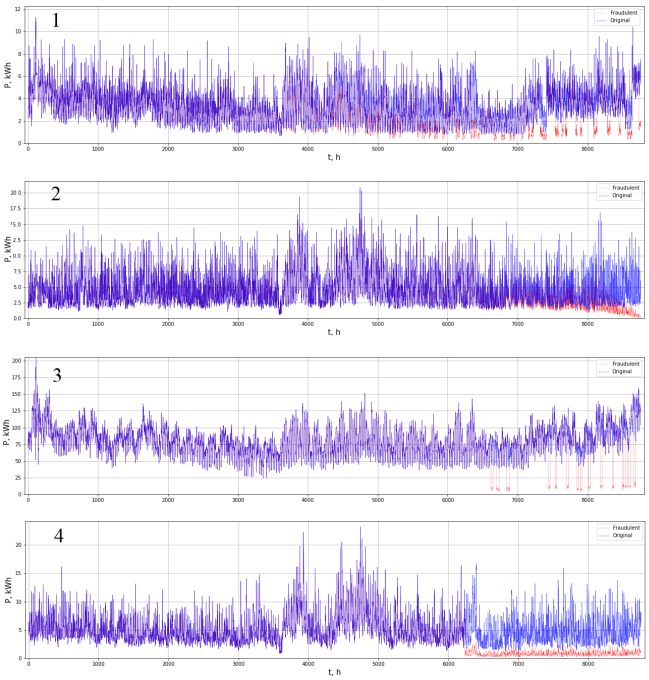


Figure 6. Attack functions: 1-BASE, 2-INTERRUPT, 3-SMART, 4-COMBINED

## IV. CONCLUSIONS

In this paper, we showed that CNN-LSTM architectures are a promising option for NTL detection. Feature extraction may be quite a laborious process. Furthermore, the hypothesis that is used as a baseline to perform feature extraction may be erroneous. From the results in comparison with others machine learning methods, we have shown that the performance of the SVM model are similarly to the CNN-LSTM for NTL simple NLT cases. The main advantage of CNNs is the fact that they are inherently good for wide range of NLT attacks. However, generally, neural networks perform better on large datasets.

TABLE II
PERFORMANCE COMPARISON OF RF, SVM, GBT, AND CNN-LSTM

| **BASE** | | | |
|---|---|---|---|
| **Models** | **Accuracy** | **Precision** | **Recall** |
| SVM | 0.95 | 0.93 | 1.0 |
| RF | 0.74 | 0.71 | 0.79 |
| GBT | 0.82 | 0.77 | 0.89 |
| CNN-LSTM | 0.95 | 0.9 | 1.0 |
| **INTERRUPT** | | | |
| **Models** | **Accuracy** | **Precision** | **Recall** |
| SVM | 0.74 | 0.86 | 0.62 |
| RF | 0.79 | 0.88 | 0.71 |
| GBT | 0.69 | 0.69 | 0.76 |
| CNN-LSTM | 0.97 | 0.94 | 1.0 |
| **SMART** | | | |
| **Models** | **Accuracy** | **Precision** | **Recall** |
| SVM | 0.95 | 0.95 | 0.95 |
| RF | 0.79 | 0.80 | 0.60 |
| GBT | 0.79 | 0.80 | 0.80 |
| CNN-LSTM | 0.95 | 0.95 | 0.95 |
| **COMBINED** | | | |
| **Models** | **Accuracy** | **Precision** | **Recall** |
| SVM | 0.66 | 0.66 | 0.63 |
| RF | 0.74 | 0.76 | 0.68 |
| GBT | 0.69 | 0.68 | 0.68 |
| CNN-LSTM | 0.87 | 0.94 | 0.81 |

Further research on alternatives combinations of CNN and LSTM architectures is needed. In addition, new open datasets with real measurment would also help the development of more consistent criteria for the NTL algorithm's selection.

## ACKNOWLEDGEMENT

This work was supported by Skoltech NGP Program (Skoltech-MIT joint project).

## REFERENCES

[1] R. Jiang, R. Lu, Y. Wang, J. Luo, C. Shen, and X. Shen, "Energy-theft detection issues for advanced metering infrastructure in smart grid," *Tsinghua Science and Technology*, vol. 19, no. 2, pp. 105–120, 2014.

[2] S. McLaughlin, B. Holbert, A. Fawaz, R. Berthier, and S. Zonouz, "A multi-sensor energy theft detection framework for advanced metering infrastructures," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1319–1330, 2013.

[3] J. L. Viegas, P. R. Esteves, R. Melício, V. Mendes, and S. M. Vieira, "Solutions for detection of non-technical losses in the electricity grid: A review," *Renewable and Sustainable Energy Reviews*, vol. 80, pp. 1256–1268, 2017.

[4] P. Glauner, J. A. Meira, P. Valtchev, R. State, and F. Bettinger, "The challenge of non-technical loss detection using artificial intelligence: A survey," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 760–775, 2017.

[5] S. S. S. R. Depuru, L. Wang, V. Devabhaktuni, and R. C. Green, "High performance computing for detection of electricity theft," *Electrical Power & Energy Systems*, vol. 47, pp. 21–30, 2013.

[6] F. B. Lewis, "Costly 'throw-ups': Electricity theft and power disruptions," *The Electricity Journal*, vol. 28, no. 7, pp. 118–135, 2015.

[7] J.-S. Chou and I. G. A. N. Yutami, "Smart meter adoption and deployment strategy for residential buildings in Indonesia," *Applied Energy*, vol. 128, pp. 336–349, 2014.

[8] G. M. Messinis and N. D. Hatziargyriou, "Review of non-technical loss detection methods," *Electric Power Systems Research*, vol. 158, pp. 250—-266, 2018.

[9] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and M. Mohamad, "Nontechnical loss detection for metered customers in power utility using support vector machines," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 1162–1171, 2010.

[10] P. Glauner, A. Boechat, L. Dolberg, R. State, F. Bettinger, Y. Rangoni, and D. Duarte, "Large-scale detection of non-technical losses in imbalanced data sets," in *2016 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2016, pp. 1–5.

[11] B. C. Costa, B. L. Alberto, A. M. Portela, W. Maduro, and E. O. Eler, "Fraud detection in electric power distribution networks using an ann-based knowledge-discovery process," *International Journal of Artificial Intelligence & Applications*, vol. 4, pp. 17–23, 2013.

[12] C. C. O. Ramos, A. N. de Souza, D. S. Gastaldello, and J. P. Papa, "Identification and feature selection of non-technical losses for industrial consumers using the software weka," in *2012 10th IEEE/IAS International Conference on Industry Applications*, 2012, pp. 1–6.

[13] P. Glauner, N. Dahringer, O. Puhachov, J. A. Meira, P. Valtchev, R. State, and D. Duarte, "Identifying irregular power usage by turning predictions into holographic spatial visualizations," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 258–265.

[14] M. N. Hasan, R. Toma, A. Nahid, M. M. M. Islam, and J. Kim, "Electricity theft detection in smart grid systems: A CNN-LSTM based approach," *Energies*, 08 2019.

[15] Z. Zheng, Y. Yang, X. Niu, H. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, 2018.

[16] S. Li, Y. Han, X. Yao, S. Yingchen, J. Wang, and Q. Zhao, "Electricity theft detection in power grids with deep learning and random forests," *Journal of Electrical and Computer Engineering*, vol. 2019, 2019.

[17] B. Kocaman and V. Tümen, "Detection of electricity theft using data processing and lstm method in distribution systems," *Sādhanā*, vol. 45, no. 1, pp. 1–10, 2020.

[18] Y.-L. Lo, S.-C. Huang, and C.-N. Lu, "Non-technical loss detection using smart distribution network measurement data," in *IEEE PES Innovative Smart Grid Technologies*, 2012, pp. 1–5.

[19] L. Chen, X. Xu, and C. Wang, "Research on anti-electricity stealing method base on state estimation," in *2011 IEEE Power Engineering and Automation Conference*, vol. 2, 2011, pp. 413–416.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[22] J. S. S. Hochreiter, "Long short-term memory," *Neural computation*, vol. 9, no. 8, p. 1735–1780, 1997.

[23] F. Bu, Y. Yuan, Z. Wang, K. Dehghanpour, and A. Kimber, "A time-series distribution test system based on real utility data," in *2019 North American Power Symposium (NAPS)*, 2019, pp. 1–6.

[24] K. M. Ghori, M. Imran, A. Nawaz, R. A. Abbasi, A. Ullah, and L. Szathmary, "Performance analysis of machine learning classifiers for non-technical loss detection," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[25] G. M. Messinis, A. E. Rigas, and N. D. Hatziargyriou, "A hybrid method for non-technical loss detection in smart distribution grids," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6080–6091, 2019.