(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 September 2020 (10.09.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/180424 A1

(51) International Patent Classification:
G06N 5/02 (2006.01) *G06F 11/00* (2006.01)
G06Q 10/06 (2012.01)

(21) International Application Number:
PCT/US2020/015698

(22) International Filing Date:
29 January 2020 (29.01.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/813,664 04 March 2019 (04.03.2019) US

(71) Applicant: **IOCURRENTS, INC.** [US/US]; 1501 4th Ave, Suite 900, Seattle, Washington 98101 (US).

(72) Inventor: **BHATTACHARYYA, Bhaskar**; c/o ioCurrents, Inc., 1501 4th Ave, Suite 900, Seattle, Washington 98101 (US).

(74) Agent: **LI, Yiheng** et al.; Seed Intellectual Property Law Group LLP, Suite 5400, 701 Fifth Avenue, Seattle, Washington 98104-7064 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,

(54) Title: DATA COMPRESSION AND COMMUNICATION USING MACHINE LEARNING

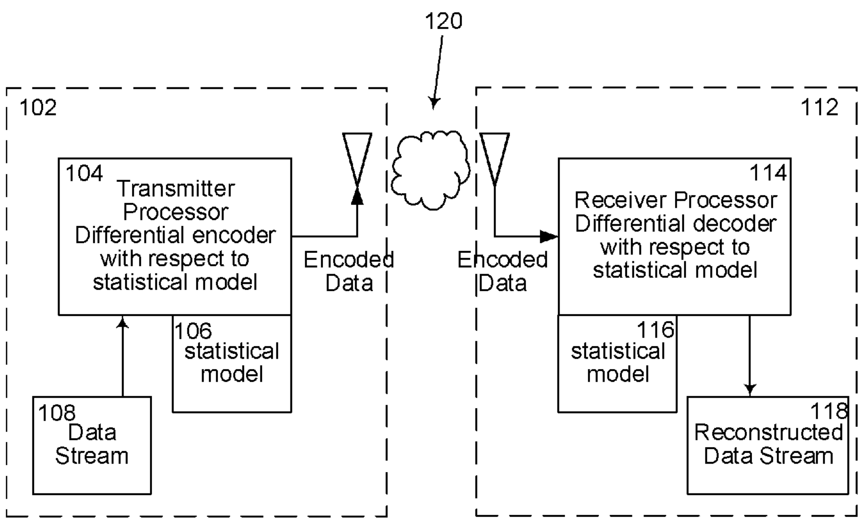

FIG. 1

(57) Abstract: A method of communicating information, comprising modeling a stream of sensor data, to produce parameters of a predictive statistical model; communicating information defining the predictive statistical model from a transmitter to a receiver; and after communicating the information defining the predictive statistical model to the receiver, communicating information characterizing subsequent sensor data from the transmitter to the receiver, dependent on an error of the subsequent sensor data with respect to a prediction of the subsequent sensor data by the statistical model. A corresponding method is also encompassed.

WO 2020/180424 A1

[Continued on next page]

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

DATA COMPRESSION AND COMMUNICATION USING MACHINE LEARNING

CROSS-REFERENCE TO RELATED APPLICATIONS

This Application claims the benefit of provisional U.S. Application No. 62/813,664, filed March 4, 2019 and entitled "SYSTEM AND METHOD FOR DATA COMPRESSION AND PRIVATE COMMUNICATION OF MACHINE DATA BETWEEN COMPUTERS USING MACHINE LEARNING," which is hereby incorporated by reference in its entirety.

BACKGROUND

Technical Field

The present disclosure relates to the field of data compression, and more particularly to lossy compression of data based on statistical properties, *e.g.*, for storage and communication of sensor data.

Description of the Related Art

In order to continuously transfer machine data time series between computers (*e.g.*, from an edge device that is collecting one or more machine's data and sending to one or more cloud servers) one computer typically transfers all of the sensor data values collected from the machine(s) at each timestamp along with timestamp data and optionally position data (*e.g.*, GPS location) or other context information, to another computer, which may be in the cloud. This communication burden is one of the main challenges in Internet of things (IoT) data transfer, due of the cost of transferring the large volume of data. Further, latency may increase and communication reliability may decrease with increasing data volume.

The process of reducing the size of a data file is often referred to as data compression. In the context of data transmission, it is called source coding; encoding done at the source of the data before it is stored or transmitted.

In signal processing, data compression, source coding, or bit-rate reduction typically involves encoding information using fewer bits than the original representation. Compression can be either lossy or lossless. Lossless compression reduces bits by identifying and eliminating redundancy. This reduction may be deterministic, *i.e.*, reduction in bits is assured, or statistical, *i.e.*, a particular type of redundancy reduction under most circumstances leads to a net reduction in bit required for encoding. No information is lost in lossless compression.

Lossless data compression algorithms usually exploit statistical redundancy to represent data without losing any information, so that the process is reversible. Lossless compression relies on the fact that real world data typically has redundancy (lack of entropy). Therefore, by reencoding the data to increase the entropy of the expression, the amount of data (bits) may be reduced. The Lempel-Ziv (LZ) compression methods employ run-length encoding. For most LZ methods, a table of previous strings is generated dynamically from earlier data in the input. The table itself is often Huffman encoded. Grammar-based codes like this can compress highly repetitive input extremely effectively, for instance, a biological data collection of the same or closely related species, a huge versioned document collection, internet archival, etc. The basic task of grammar-based codes is constructing a context-free grammar deriving a single string. Other practical grammar compression algorithms include Sequitur and Re-Pair.

Some lossless compressors use probabilistic models, such as prediction by partial matching. The Burrows-Wheeler transform can also be viewed as an indirect form of statistical modeling. In a further refinement of the direct use of probabilistic modeling, statistical estimates can be coupled to an algorithm called arithmetic coding, which uses the mathematical calculations of a finite-state machine to produce a string of encoded bits from a series of input data symbols. It uses an internal memory state to avoid the need to perform a one-to-one mapping of individual input symbols to distinct representations that use an integer number of bits, and it clears out the internal memory only after encoding the entire string of data symbols. Arithmetic coding applies especially well to adaptive data compression tasks where the statistics vary and are context-dependent, as it can be easily coupled with an adaptive model of the probability distribution of the input data.

Lossy compression typically reduces the number of bits by removing unnecessary or less important information. This can involve predicting which signal aspects may be considered noise, and/or which signal aspects have low importance for the ultimate use of the data. Lossy data compression is, in one aspect, the converse of lossless data compression, which loses information. However, subject to loss of information, the techniques of lossless compression may also be employed with lossy data compression.

There is a close connection between machine learning and compression: a system that predicts the posterior probabilities of a sequence given its entire history can be used for optimal data compression (by using arithmetic coding on the output distribution) while an optimal compressor can be used for prediction (by finding the symbol that compresses best, given the previous history).

Compression algorithms can implicitly map strings into implicit feature space vectors, and compression-based similarity measures used to compute similarity within these feature spaces. For each compressor C(.) we define an associated vector space ℵ, such that C(.) maps an input string x, corresponds to the vector norm ||~x||.

In lossless compression, and typically lossy compression as well, information redundancy is reduced, using methods such as coding, pattern recognition, and linear prediction to reduce the amount of information used to represent the uncompressed data. Due to the nature of lossy algorithms, quality suffers when a file is decompressed and recompressed (digital generation loss). (Lossless compression may be achieved through loss of non-redundant information, so increase in entropy is not assured.)

In lossy compression, the lost information is, or is treated as, noise. One way to filter noise is to transform the data to a representation where the supposed signal is concentrated in regions of the data space, to form a sparse distribution. The sparse regions of the distribution may be truncated, *e.g.*, by applying a threshold, and the remaining dense regions of the distribution may be further transformed or encoded.

Multiple different methods may be employed, to reduce noise based on different criteria.

See, 10003794; 10028706; 10032309; 10063861; 10091512; 5243546; 5486762; 5515477; 5561421; 5659362; 6081211; 6219457; 6223162; 6300888; 6356363; 6362756; 6389389; 6404925; 6404932; 6490373; 6510250; 6606037; 6664902; 6671414; 6675185; 6678423; 6751354; 6757439; 6760480; 6774917; 6795506; 6801668; 6832006; 6839003; 6895101; 6895121; 6927710; 6941019; 7006568; 7050646; 7068641; 7099523; 7126500; 7146053; 7246314; 7266661; 7298925; 7336720; 7474805; 7483871; 7504970; 7518538; 7532763; 7538697; 7564383; 7578793; 7605721; 7612692; 7629901; 7630563; 7645984; 7646814; 7660295; 7660355; 7719448; 7743309; 7821426; 7881544; 7885988; 7936932; 7961959; 7961960; 7970216; 7974478; 8005140; 8017908; 8112624; 8160136; 8175403; 8178834; 8185316; 8204224; 8238290; 8270745; 8306340; 8331441; 8374451; 8411742; 8458457; 8480110; 8509555; 8540644; 8644171; 8694474; 8718140; 8731052; 8766172; 8964727; 9035807; 9111333; 9179147; 9179161; 9339202; 9478224; 9492096; 9705526; 9812136; 9940942; 20010024525; 20010031089; 20020028021; 20020076115; 20020090139; 20020131084; 20020175921; 20020176633; 20030018647; 20030059121; 20030086621; 20030098804; 20040001543; 20040001611; 20040015525; 20040027259; 20040085233; 20040165527; 20040221237; 20050069224; 20050147172; 20050147173; 20050276323; 20060053004; 20060061795; 20060111635; 20060143454; 20060165163; 20060200709; 20070083491; 20070216545; 20070217506; 20070223582; 20070278395; 20070297394; 20080031545; 20080037880; 20080050025; 20080050026; 20080050027; 20080050029; 20080050047; 20080055121; 20080126378; 20080152235; 20080154928; 20080189545; 20090041021; 20090138715; 20090140893; 20090140894; 20090212981; 20090232408; 20090234200; 20090262929; 20090284399; 20090289820; 20090292475; 20090294645; 20090322570; 20100114581; 20100187414; 20100202442; 20110019737; 20110032983; 20110176606; 20110182524; 20110200266; 20110263967; 20110299455; 20120014435;

20120051434; 20120069895; 20120143510; 20120259557; 20130013574; 20130080073; 20130289424; 20140010288; 20140025342; 20140184430; 20140303944; 20140307770; 20140370836; 20140376827; 20150086013; 20150100244; 20150341643; 20150381994; 20160042744; 20160055855; 20160256112; 20160261997; 20160292589; 20160372123; 20170046615; 20170105004; 20170105005; 20170310972; 20170310974; 20170337711; 20170359584; 20180124407; 20180176556; 20180176563; 20180176582; 20180211677; 20180293778; and 20180295375.

Wireless Sensor Networks (WSN) typically consist of a large number of sensors distributed in a sensing area to serve different tasks, such as continuous environmental monitoring. These networks are intended to continuously sense an area of interest and transmit the sensed data to a sink node. Due to the power consumption constraints, it is inefficient to directly transmit the raw sensed data to the sink, as they often exhibit a high correlation in the spatial and temporal domains and can be efficiently compressed to reduce power and bandwidth requirements, and reduce latency, and provide greater opportunity for error detection and correction (EDC) encoding. See:

10004183; 10006779; 10007592; 10008052; 10009063; 10009067; 10010703; 10020844; 10024187; 10027397; 10027398; 10032123; 10033108; 10035609; 10038765; 10043527; 10044409; 10046779; 10050697; 10051403; 10051630; 10051663; 10063280; 10068467; 10069185; 10069535; 10069547; 10070321; 10070381; 10079661; 10084223; 10084868; 10085425; 10085697; 10089716; 10090594; 10090606; 10091017; 10091787; 10103422; 10103801; 10111169; 10116697; 10121338; 10121339; 10122218; 10133989; 10135145; 10135146; 10135147; 10135499; 10136434; 10137288; 10139820; 10141622; 10142010; 10142086; 10144036; 10148016; 10149129; 10149131; 10153823; 10153892; 10154326; 10155651; 10168695; 10170840; 10171501; 10178445; 10187850; 10194437; 10200752; 6735630; 6795786; 6826607; 6832251; 6859831; 7020701; 7081693; 7170201; 7207041; 7231180; 7256505; 7328625; 7339957; 7361998; 7365455; 7385503; 7398164; 7429805; 7443509; 7487066; 7605485;

7609838; 7630736; 7660203; 7671480; 7710455; 7719416; 7764958; 7788970; 7797367; 7802015; 7805405; 7844687; 7873673; 7881206; 7908928; 7953559; 7957222; 7990262; 7996342; 8000314; 8010319; 8011255; 8013731; 8013732; 8024980; 8026113; 8026808; 8031650; 8035511; 8044812; 8064412; 8073331; 8086864; 8098485; 8104993; 8111156; 8112381; 8140658; 8171136; 8193929; 8193930; 8194655; 8194858; 8195814; 8199635; 8212667; 8214082; 8214370; 8219848; 8221273; 8223010; 8225129; 8233471; 8260575; 8264401; 8265657; 8279067; 8279080; 8280671; 8282517; 8289184; 8305899; 8325030; 8330596; 8335304; 8350750; 8359347; 8370935; 8373576; 8375442; 8379564; 8395496; 8410931; 8417762; 8421274; 8446884; 8451766; 8489063; 8493601; 8529383; 8533473; 8536998; 8544089; 8552861; 8559271; 8572290; 8582481; 8585517; 8585606; 8600560; 8615374; 8625496; 8630965; 8635654; 8638217; 8660786; 8666357; 8687810; 8688850; 8700064; 8704656; 8711743; 8733168; 8756173; 8776062; 8781768; 8787246; 8795172; 8805579; 8810429; 8812007; 8812654; 8816850; 8822924; 8832244; 8836503; 8855011; 8855245; 8867309; 8867310; 8873335; 8873336; 8879356; 8885441; 8892624; 8892704; 8922065; 8923144; 8924587; 8924588; 8930571; 8949989; 8954377; 8964708; 8971432; 8982856; 8983793; 8987973; 8990032; 8994551; 9004320; 9017255; 9026273; 9026279; 9026336; 9028404; 9032058; 9063165; 9065699; 9072114; 9074731; 9075146; 9090339; 9103920; 9105181; 9111240; 9115989; 9119019; 9129497; 9130651; 9141215; 9148849; 9152146; 9154263; 9164292; 9191037; 9202051; 9210436; 9210938; 9226304; 9232407; 9233466; 9239215; 9240955; 9282029; 9288743; 9297915; 9305275; 9311808; 9325396; 9356776; 9363175; 9372213; 9374677; 9386522; 9386553; 9387940; 9397795; 9398576; 9402245; 9413571; 9417331; 9429661; 9430936; 9439126; 9445445; 9455763; 9459360; 9470809; 9470818; 9492086; 9495860; 9500757; 9515691; 9529210; 9571582; 9576404; 9576694; 9583967; 9584193; 9585620; 9590772; 9605857; 9608740; 9609810; 9615226; 9615269; 9615792; 9621959; 9628165; 9628286; 9628365; 9632746; 9639100; 9640850; 9651400; 9656389; 9661205; 9662392; 9666042; 9667317; 9667653; 9674711; 9681807; 9685992; 9691263; 9699768; 9699785; 9701325; 9705561;

9705610; 9711038; 9721210; 9722318; 9727115; 9728063; 9729197; 9730160; 9735833; 9742462; 9742521; 9743370; 9746452; 9748626; 9749013; 9749053; 9749083; 9753022; 9753164; 9762289; 9766320; 9766619; 9768833; 9769020; 9769128; 9769522; 9772612; 9776725; 9780834; 9781700; 9787412; 9788326; 9788354; 9791910; 9793951; 9793954; 9793955; 9800327; 9806818; 9812754; 9816373; 9816897; 9820146; 9824578; 9831912; 9838078; 9838736; 9838760; 9838896; 9846479; 9847566; 9847850; 9853342; 9854551; 9854994; 9858681; 9860075; 9860820; 9863222; 9865911; 9866276; 9866306; 9866309; 9871282; 9871283; 9871558; 9874923; 9876264; 9876570; 9876571; 9876587; 9876605; 9878138; 9878139; 9882257; 9884281; 9887447; 9888081; 9891883; 9893795; 9894852; 9896215; 9900177; 9902499; 9904535; 9906269; 9911020; 9912027; 9912033; 9912381; 9912382; 9912419; 9913006; 9913139; 9917341; 9927512; 9927517; 9929755; 9930668; 9931036; 9931037; 9935703; 9946571; 9948333; 9948354; 9948355; 9948477; 9953448; 9954286; 9954287; 9957052; 9960808; 9960980; 9965813; 9967002; 9967173; 9969329; 9970993; 9973299; 9973416; 9973940; 9974018; 9980223; 9983011; 9990818; 9991580; 9997819; 9998870; 9998932; 9999038; 20030107488; 20030151513; 20040083833; 20040090329; 20040090345; 20040100394; 20040128097; 20040139110; 20050017602; 20050090936; 20050210340; 20050213548; 20060026017; 20060029060; 20060175606; 20060206246; 20060243055; 20060243056; 20060243180; 20070038346; 20070090996; 20070101382; 20070195808; 20070210916; 20070210929; 20070221125; 20070224712; 20070239862; 20080031213; 20080074254; 20080122938; 20080129495; 20080215609; 20080219094; 20080253283; 20080256166; 20080256167; 20080256253; 20080256384; 20080256548; 20080256549; 20080309481; 20090007706; 20090009317; 20090009323; 20090009339; 20090009340; 20090058088; 20090058639; 20090059827; 20090070767; 20090146833; 20090149722; 20090161581; 20090168653; 20090196206; 20090198374; 20090210173; 20090210363; 20090296670; 20090303042; 20090322510; 20100031052; 20100039933; 20100054307; 20100074054; 20100100338; 20100109853; 20100125641;

20100148940; 20100152619; 20100152909; 20100176939; 20100201516; 20100211787; 20100254312; 20100278060; 20100312128; 20110035271; 20110035491; 20110045818; 20110101788; 20110137472; 20110158806; 20110176469; 20110191496; 20110248846; 20110293278; 20110310779; 20120014289; 20120089370; 20120092155; 20120106397; 20120123284; 20120127020; 20120127924; 20120173171; 20120178486; 20120190386; 20120215348; 20120218376; 20120250863; 20120257530; 20120262291; 20120265716; 20130016625; 20130016636; 20130041627; 20130044183; 20130046463; 20130048436; 20130076531; 20130076532; 20130078912; 20130097276; 20130107041; 20130113631; 20130148713; 20130153060; 20130155952; 20130176872; 20130180336; 20130201316; 20130207815; 20130244121; 20130258904; 20130265874; 20130265915; 20130265981; 20130314273; 20130320212; 20130332010; 20130332011; 20130332025; 20140010047; 20140062212; 20140114549; 20140124621; 20140153674; 20140191875; 20140192689; 20140216144; 20140225603; 20140253733; 20140263418; 20140263430; 20140263989; 20140264047; 20140266776; 20140266785; 20140268601; 20140273821; 20140275849; 20140299783; 20140301217; 20140312242; 20140349597; 20140350722; 20140355499; 20140358442; 20150046582; 20150049650; 20150078738; 20150081247; 20150082754; 20150094618; 20150119079; 20150139425; 20150164408; 20150178620; 20150192682; 20150249486; 20150268355; 20150280863; 20150286933; 20150288604; 20150294431; 20150316926; 20150330869; 20150338525; 20150343144; 20150351084; 20150351336; 20150363981; 20160000045; 20160012465; 20160025514; 20160044035; 20160051791; 20160051806; 20160072547; 20160081551; 20160081586; 20160082589; 20160088517; 20160091730; 20160100444; 20160100445; 20160152252; 20160173959; 20160174148; 20160183799; 20160189381; 20160202755; 20160260302; 20160260303; 20160300183; 20160314055; 20160323839; 20160323841; 20160338617; 20160338644; 20160345260; 20160353294; 20160356665; 20160356666; 20160378427; 20170006140; 20170013533;

20170021204; 20170072851; 20170078400; 20170106178; 20170116383; 20170126332; 20170135041; 20170151964; 20170167287; 20170169912; 20170171807; 20170171889; 20170172472; 20170172473; 20170173262; 20170177435; 20170177542; 20170180214; 20170181098; 20170181628; 20170183243; 20170195823; 20170201297; 20170213345; 20170217018; 20170222753; 20170223653; 20170228998; 20170259050; 20170259942; 20170264805; 20170268954; 20170276655; 20170281092; 20170284839; 20170287522; 20170289323; 20170289812; 20170295503; 20170296104; 20170302756; 20170330431; 20170331899; 20170346609; 20170347297; 20170353865; 20170374619; 20180017392; 20180019862; 20180024029; 20180034912; 20180039316; 20180049638; 20180058202; 20180077663; 20180078747; 20180078748; 20180124181; 20180129902; 20180132720; 20180148180; 20180148182; 20180162549; 20180164439; 20180166962; 20180170575; 20180181910; 20180182116; 20180212787; 20180213348; 20180222388; 20180246696; 20180271980; 20180278693; 20180278694; 20180293538; 20180310529; 20180317140; 20180317794; 20180326173; 20180338017; 20180338282; 20180343304; 20180343482; 20180375940; 20190014587; 20190015622; 20190020530; 20190036801; and 20190037558.

Spatial correlation in WSN refers to, *e.g.*, the correlation between the sensed data at spatially adjacent sensor nodes. On the other hand, temporal correlation usually refers to the slow varying nature of the sensed data. Compressive sensing (CS) is a tool that provides a means to process and transport correlated data in an efficient manner by exploring the sparsity of these data. Temporal correlation can be modeled in the form of a multiple measurement vector (MMV), where it models the source as an auto regressive (AR) process and then incorporates such information into the framework of sparse Bayesian learning for sparse signal recovery and converts MMV to block single measurement vector (SMV) model. Compressive sensing theory provides an elegant mathematical framework to compress and recover signals using a small number of linear measurements. Under certain conditions on the measurement matrix, the acquired signal can be perfectly reconstructed from these measurements.

A mean is a commonly used measure of central tendency, and is influenced by every value in a sample according to the formula:

$$\mu = \frac{\sum X}{N}$$

where μ is population mean, and $\overline{X}$ is sample mean.

A standard deviation is a measure of variability, according to the formula:

$$\sigma = \sqrt{\frac{\sum (X - \mu)^2}{N}}$$

(if μ is unknown, use $\overline{X}$)

A small sample bias may be corrected by dividing by n-1, where n is the number of samples, *i.e.*:

$$\sigma = \sqrt{\frac{\sum (X - \overline{X})^2}{n - 1}}$$

A normal distribution has a bell shaped curve, and is a reasonably accurate description of many (but not all) natural distributions introduced by a random process. It is unimodal, symmetrical, has points of inflection at $\mu \pm \sigma$, has tails that approach x-axis, and is completely defined by its mean and standard deviation.

The standard error of the mean, is a standard deviation of sampling error of different samples of a given sample size. For a sampling error of ($\overline{X}$ - μ), as n increases, variability decreases:

$$\sigma_{\overline{X}} = \frac{\sigma}{\sqrt{n}}$$

"File Compression Possibilities". A Brief guide to compress a file in 4 different ways.

"Intel labs berkeley data," www.select.cs.cmu.edu/data/labapp3/.

Alwakeel, Ahmed S., Mohamed F. Abdelkader, Karim G. Seddik, and Atef Ghuniem. "Exploiting temporal correlation of sparse signals in wireless sensor networks.” In Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th, pp. 1-6. IEEE, 2014.

Arcangel, Cory. “On Compression” (2013)

Baraniuk, R. G., “Compressive sensing [lecture notes],” *IEEE, Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

Ben-Gal, I. (2008). “On the Use of Data Compression Measures to Analyze Robust Designs”, 54 (3). IEEE Transactions on Reliability: 381–388.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. 2011, 3, 1–122.

Cai, J.F.; Candes, E.J.; Shen, Z.W. A singular value thresholding algorithm for matrix completion. SIAM J. Optim. 2010, 20, 1956–1982.

Caione, C.; Brunelli, D.; Benini, L. Distributed compressive sampling for lifetime optimization in dense wireless sensor networks. IEEE Trans. Ind. Inf. 2012, 8, 30–40.

Candes, E. J., M. B. Wakin, and S. P. Boyd, “Enhancing sparsity by reweighted l 1 minimization,” *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

Candes, E.; Recht, B. Exact matrix completion via convex optimization. Commun. ACM 2012, 55, 111–119.

Candes, E.J.; Recht, B. Exact matrix completion via convex optimization. Found. Comput. Math. 2009, 9, 717–772.

Candes, E.J.; Romberg, J.; Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory 2006, 52, 489–509.

CCITT Study Group VIII und die Joint Photographic Experts Group (JPEG) von ISO/IEC Joint Technical Committee 1/Subcommittee 29/Working Group 10 (1993), “Annex D – Arithmetic coding”, in ITU-T (in German), Recommendation T.81: Digital Compression and Coding of Continuous-tone Still images – Requirements and guidelines, pp. 54 ff.

Cevher, V., A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 155–168.

Chanda P, Bader JS, Elhaik E; Elhaik; Bader (27 Jul 2012). "HapZipper: sharing HapMap populations just got easier", Nucleic Acids Research. 40 (20): e159. doi:10.1093/nar/gks709. PMC 3488212. PMID 22844100.

Charbiwala, Z., Y. Kim, S. Zahedi, J. Friedman, and M. B. Srivastava, "Energy efficient sampling for event detection in wireless sensor networks," in *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design*. ACM, 2009, pp. 419–424.

Cheng, J.; Ye, Q.; Jiang, H.; Wang, D.; Wang, C. STCDG: An efficient data gathering algorithm based on matrix completion for wireless sensor networks. IEEE Trans. Wirel. Commun. 2013, 12, 850–861.

Christley S, Lu Y, Li C, Xie X; Lu; Li; Xie (Jan 15, 2009). "Human genomes as email attachments". Bioinformatics. 25 (2): 274–5. doi:10.1093/bioinformatics/btn582. PMID 18996942.

Claude Elwood Shannon (1948), Alcatel-Lucent, ed., "A Mathematical Theory of Communication" (in German), Bell System Technical Journal 27 (3–4)

Cliff Reader (2016-08-31), Society of Photo-Optical Instrumentation Engineers, ed., [Vortragsmitschnitt, ab 3:05:10 "Patent landscape for royalty-free video coding"], Applications of Digital Image Processing XXXIX (San Diego, California)

Coalson, Josh. "FLAC Comparison".

Donoho, D. L., "Compressed sensing," *IEEE Transactions on, Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

Donoho, D.L. Compressed sensing. IEEE Trans. Inf. Theory 2006, 52, 1289–1306.

en.wikipedia.org/wiki/Data_compression

Faxin Yu; Hao Luo; Zheming Lu (2010). Three-Dimensional Model Analysis and Processing. Berlin: Springer. p. 47. ISBN 9783642126512.

Gleichman, S.; Eldar, Y.C. Blind compressed sensing. IEEE Trans. Inf. Theory 2011, 57, 6958–6975.

Goel, S., and T. Imielinski, "Prediction-based monitoring in sensor networks: taking lessons from mpeg," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 5, pp. 82–98, 2001.

Goldstein, T.; O'Donoghue, B.; Setzer, S.; Baraniuk, R. Fast alternating direction optimization methods. SIAM J. Imaging Sci. 2014, 7, 1588–1623.

Golub, G.H.; Van Loan, C.F. Matrix Computations; JHU Press: Baltimore, MD, USA, 2012.

Graphics & Media Lab Video Group (2007). Lossless Video Codecs Comparison. Moscow State University.

Grimes, C. A., "Design of a wireless sensor network for long-term, in-situ monitoring of an aqueous environment," *Sensors*, vol. 2, no. 11, pp. 455–472, 2002.

He, B.; Tao, M.; Yuan, X. Alternating direction method with Gaussian back substitution for separable convex programming. SIAM J. Optim. 2012, 22, 313–340.

Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Siences, Maui, HI, USA, 4–7 January 2000; p. 223.

Hilbert, Martin; López, Priscila (1 April 2011). "The World's Technological Capacity to Store, Communicate, and Compute Information". Science. 332 (6025): 60–65. Bibcode:2011Sci...332...60H. doi:10.1126/science.1200970. PMID 21310967.

Hu, Y.; Zhang, D.; Ye, J.; Li, X.; He, X. Fast and accurate matrix completion via truncated nuclear norm regularization. IEEE Trans. Pattern Anal. Mach. Intell. 2013, 35, 2117–2130.

Huffman, David Albert (1952-09), "A method for the construction of minimum-redundancy codes" (in German), Proceedings of the IRE 40 (9): pp. 1098–1101, doi:10.1109/JRPROC.1952.273898

Jaiswal, R.C. (2009). Audio-Video Engineering. Pune, Maharashtra: Nirali Prakashan. p. 3.41. ISBN 9788190639675.

Kadkhodaie, M.; Christakopoulou, K.; Sanjabi, M.; Banerjee, A. Accelerated alternating direction method of multipliers. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 497–506.

Kong, L.; Xia, M.; Liu, X.Y.; Chen, G.; Gu, Y.; Wu, M.Y.; Liu, X. Data loss and reconstruction in wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. 2014, 25, 2818–2828.

Korn, D.; et al. “RFC 3284: The VCDIFF Generic Differencing and Compression Data Format”. Internet Engineering Task Force. (2002).

Korn, D.G.; Vo, K.P. (1995), B. Krishnamurthy, ed., Vdelta: Differencing and Compression, Practical Reusable Unix Software, New York: John Wiley & Sons, Inc.

Lachowski, R.; Pellenz, M.E.; Penna, M.C.; Jamhour, E.; Souza, R.D. An efficient distributed algorithm for constructing spanning trees in wireless sensor networks. Sensors 2015, 15, 1518–1536.

Lane, Tom. “JPEG Image Compression FAQ, Part 1”. Internet FAQ Archives. Independent JPEG Group.

Larsen, R.M. PROPACK-Software for Large and Sparse SVD Calculations. Available online: sun.stanford.edu/~rmunk/PROPACK.

Li, S.X.; Gao, F.; Ge, G.N.; Zhang, S.Y. Deterministic construction of compressed sensing matrices via algebraic curves. IEEE Trans. Inf. Theory 2012, 58, 5035–5041.

Liu, X.Y.; Zhu, Y.; Kong, L.; Liu, C.; Gu, Y.; Vasilakos, A.V.; Wu, M.Y. CDC: Compressive data collection for wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. 2015, 26, 2188–2197.

Liu, Y.; He, Y.; Li, M.; Wang, J.; Liu, K.; Li, X. Does wireless sensor network scale? A measurement study on GreenOrbs. IEEE Trans. Parallel Distrib. Syst. 2013, 24, 1983–1993.

Luo, C., F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," *ACM, Proceedings of the 15th annual international conference on Mobile computing and networking*, pp. 145–156, 2009.

Luo, C., F. Wu, J. Sun, and C. W. Chen, "Efficient measurement generation and pervasive sparsity for compressive data gathering," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 12, pp. 3728–3738, 2010.

Luo, C.; Wu, F.; Sun, J.; Chen, C.W. Compressive data gathering for large-scale wireless sensor networks. In Proceedings of the 15th ACM International Conference on Mobile Computing and Networking, Beijing, China, 20–25 September 2009; pp. 145–156.

M. Hosseini, D. Pratas, and A. Pinho. 2016. A survey on data compression methods for biological sequences. Information 7(4):(2016): 56

Mahdi, O.A.; Mohammed, M.A.; Mohamed, A.J. (November 2012). "Implementing a Novel Approach an Convert Audio Compression to Text Coding via Hybrid Technique". International Journal of Computer Science Issues. 9 (6, No. 3): 53–59.

Mahmud, Salauddin (March 2012). "An Improved Data Compression Method for General Data". International Journal of Scientific & Engineering Research. 3(3):2.

Mahoney, Matt. "Rationale for a Large Text Compression Benchmark". Florida Institute of Technology. (2006) cs.fit.edu/mmahoney/compression/rationale.htm

Marak, Laszlo. "On image compression" University of Marne la Vallee (2013).

Mittal, S.; Vetter, J. (2015), "A Survey Of Architectural Approaches for Data Compression in Cache and Main Memory Systems", IEEE Transactions on Parallel and Distributed Systems, IEEE

Nasir Ahmed, T. Natarajan, Kamisetty Ramamohan Rao (1974-01), "Discrete Cosine Transform" (in German), IEEE Transactions on Computers C-23 (1): pp. 90–93, doi:10.1109/T-C.1974.223784

Navqi, Saud; Naqvi, R.; Riaz, R.A.; Siddiqui, F. (April 2011). "Optimized RTL design and implementation of LZW algorithm for high bandwidth applications" Electrical Review. 2011 (4): 279–285.

Pavlichin DS, Weissman T, Yona G; Weissman; Yona (September 2013). "The human genome contracts again". Bioinformatics. 29 (17): 2199–202. doi:10.1093/bioinformatics/btt362. PMID 23793748.

Pham, N.D.; Le, T.D.; Park, K.; Choo, H. SCCS: Spatiotemporal clustering and compressing schemes for efficient data collection applications in WSNs. Int. J. Commun. Syst. 2010, 23, 1311–1333.

Pujar, J.H.; Kadlaskar, L.M. (May 2010). "A New Lossless Method of Image Compression and Decompression Using Huffman Coding Techniques" Journal of Theoretical and Applied Information Technology. 15 (1): 18–23.

Roughan, M.; Zhang, Y.; Willinger, W.; Qiu, L.L. Spatio-temporal compressive sensing and internet traffic matrices. IEEE ACM Trans. Netw. 2012, 20, 662–676.

Salomon, David (2008). A Concise Introduction to Data Compression. Berlin: Springer. ISBN 9781848000728.

Scully, D.; Carla E. Brodley (2006). "Compression and machine learning: A new perspective on feature space vectors" Data Compression Conference, 2006.

Shmilovici A.; Kahiri Y.; Ben-Gal I.; Hauser S. (2009). "Measuring the Efficiency of the Intraday Forex Market with a Universal Data Compression Algorithm" 33(2). Computational Economics: 131–154.

Shuman, D.I.; Narang, S.K.; Frossard, P.; Ortega, A.; Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag. 2013, 30, 83–98.

Silberstein, A., R. Braynard, and J. Yang, "Constraint chaining: on energy-efficient continuous monitoring in sensor networks," in *Proceed ings of the*

2006 ACM SIGMOD international conference on Management of data. ACM, 2006, pp. 157–168.

Sullivan, G. J.; Ohm, J.-R.; Han, W.-J.; Wiegand, T., (December 2012). “Overview of the High Efficiency Video Coding (HEVC) Standard” IEEE Transactions on Circuits and Systems for Video Technology. IEEE. 22 (12).

Tank, M.K. (2011). Implementation of Limpel-Ziv algorithm for lossless compression using VHDL. Thinkquest 2010: Proceedings of the First International Conference on Contours of Computing Technology. Berlin: Springer. pp. 275–283.

The Olympus WS-120 digital speech recorder, according to its manual, can store about 178 hours of speech-quality audio in .WMA format in 500 MB of flash memory.

Toh, K.C.; Yun, S. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pac. J. Optim. 2010, 6, 615–640.

US patent 2,605,361, C. Chapin Cutler, “Differential Quantization of Communication Signals”, issued 1952-07-29

Wade, Graham (1994). Signal coding and processing (2 ed.). Cambridge University Press. p. 34. ISBN 978-0-521-42336-6. “The broad objective of source coding is to exploit or remove ‘inefficient’ redundancy in the PCM source and thereby achieve a reduction in the overall source rate R.”

Wang, Donghao, Wan, Jiangwen, Nie, Zhipeng, Zhang, Qiang, and Fei, Zhijie, “Efficient Data Gathering Methods in Wireless Sensor Networks Using GBTR Matrix Completion”, *Sensors* 2016, *16*(9), 1532; doi:10.3390/s1601532

William K. Pratt, Julius Kane, Harry C. Andrews: “Hadamard transform image coding”, in Proceedings of the IEEE 57.1 (1969): Seiten 58–68

Wolfram, Stephen (2002). A New Kind of Science. Wolfram Media, Inc. p. 1069. ISBN 1-57955-008-8.

Xiang, L., J. Luo, C. Deng, A. V. Vasilakos, and W. Lin, "Dual-level compressed aggregation: Recovering fields of physical quantities from incomplete sensory data," *arXiv preprint arXiv:1107.4873*, 2011.

Xiang, L.; Luo, J.; Rosenberg, C. Compressed data aggregation: Energy-efficient and high-fidelity data collection. IEEE ACM Trans. Netw. 2013, 21, 1722–1735.

Yang, X., K. G. Ong, W. R. Dreschel, K. Zeng, C. S. Mungle, and

Yoon, S.; Shahabi, C. The Clustered AGgregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks. ACM Trans. Sens. Netw. 2007, 3, 3.

Zhang Z., and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, pp. 912–926, 2011.

Zheng, H., S. Xiao, X. Wang, and X. Tian, "Energy and latency analysis for in-network computation with compressive sensing in wireless sensor networks," *INFOCOM*, pp. 2811–2815, 2012.

Zwicker, Eberhard; et al. (1967). The Ear As A Communication Receiver. Melville, NY: Acoustical Society of America.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a block diagram of a system including a transmitter and a receiver according to some embodiments of the presently disclosed technology.

Figure 2 shows a flowchart of actions performed by the transmitter and the receiver according to some embodiments of the presently disclosed technology.

DETAILED DESCRIPTION

The present disclosure concerns communicating sensor data. In accordance with some embodiments, the technique(s) disclosed significantly compresses the data volume by using a common machine learning based model on both send and receive sides, and sending only independent sensor variables and discrete standard error values of dependent sensor variables based on the prediction from the generated model instead of sending all the sensor data as continuous variables. Thus, the presently disclosed technology reduces data volume at the expense of loss of precision. The loss of precision can be designed carefully such that it serves the intended purpose of the data, *e.g.*, human viewing. In some embodiments, various and applicable lossless data compression techniques (e.g., Huffman Encoding) can be implemented before, after, and/or otherwise in combination with the presently disclosed lossy compression technology. For example, after applying the presently disclosed technology, the independent parameter(s) (e.g., independent sensor variables) and/or contextual data (e.g., timestamps, latitudes, longitudes, or the like) can be compressed using other compression techniques before data transmission.

Consider a system where one or multiple machines are connected to an edge device. At the start of the system, the transmitting device (*e.g.*, an edge computer) must transfer all of the machine data to the receiving device (*e.g.*, a cloud server). When enough data are transmitted, both sides of the system generate an identical machine learning based model. Once the model generation is complete on both sides, the system synchronously switches to a reduced transmission mode, sending only computed error values, *e.g.*, standard error values, as the dependent sensor variables' data.

Over time, the models may be updated; however, this updating must occur on the edge device due to the loss of precision introduced in compression. New models may be generated as needed and sent over a high bandwidth and/or cheap communication channel (*e.g.*, LAN, WLAN, or cellular communication) when available, whereas lower data rate and/or expensive communication media (*e.g.*, satellite communication, LoRaWAN, etc.) can be used for sending machine data. The model synchronization process may be scheduled for a period when the edge device has access to a high bandwidth and/or cheap communication medium (*e.g.*, when a vehicle with a deployed edge device enters a certain geographic area). The system cannot start using the new model until both sender and receiver have synchronized the new model and new training error statistics at which point both sides must switch synchronously and begin sending and receiving compressed data according to the updated compression mechanism.

Due to the potentially large size of a machine learning based model, the model may be stored as a database lookup table, reducing the model size considerably at the expense of loss in precision. The model data rows may be restricted to the practical possible combinations of input independent variables and hence shrink the model's size. A typical model saved in table form and including a diesel engine's speed (*i.e.*, Revolutions Per Minute) from 0 to 2000 and engine load 0 to 100%, will have 200001 rows (*i.e.*, 2000 × 100 rows + one row for engine speed and engine load percent both zero). Thus, a 20 sensor model (2 independent and 18 dependent) would require around 16MB space considering 4 bytes of storage per sensor.

In some embodiments, the edge device runs a machine learning based method on a training dataset collected over time from a machine and generate a model that represents the relationships between independent and dependent variables. Once the model is built, it would generate the error statistics (*i.e.*, mean training error and standard deviation of training errors) for the training period from the difference between model predicted dependent sensor values and actual measured dependent sensor values, and save the sensor specific error statistics. Once the ML based model is built using training data and the error means and error standard deviations of dependent sensors are generated and stored on both sender and receiver side, at run time the edge device can measure all the independent and dependent sensor variables and compute the standard errors of all dependent sensor values from the difference between measured dependent sensor values and predicted sensor values and error mean and error standard deviations, and transmit only the standard errors of dependent sensor values. The receiving computer can generate the same model independently from the exact same data it received from edge before. When the receiving computer receives the standard error values for each sensor, it can compute the actual sensor data values back from the standard error values, using model predicted sensor value for the specific independent sensor variables and training error statistics.

It is therefore an object to provide a method of communicating information, comprising: modeling a stream of sensor data, to produce parameters of a predictive statistical model; communicating information defining the predictive statistical model from a transmitter to a receiver; and after communicating the information defining the predictive statistical model to the receiver, communicating information characterizing subsequent sensor data from the transmitter to the receiver, dependent on an error of the subsequent sensor data with respect to a prediction of the subsequent sensor data by the statistical model.

It is also an object to provide a method of synchronizing a state of a transmitter and a receiver, to communicate a stream of sensor data, comprising: modeling the stream of sensor data input to the transmitter, to produce parameters of a predictive statistical model; communicating information defining the predictive statistical model to the receiver; and communicating information characterizing subsequent sensor data from the transmitter to the receiver, as a statistically normalized differential encoding of the subsequent sensor data with respect to a prediction of the subsequent sensor data by the predictive statistical model.

It is a further object to provide a system for receiving communicated information, comprising: a predictive statistical model, stored in a memory, derived by modeling a stream of sensor data; a communication port configured to receive a communication from a transmitter; and at least one processor, configured to: receive information defining the predictive statistical model from the transmitter; and after reception of the information defining the predictive statistical model, receive information characterizing subsequent sensor data from the transmitter, dependent on an error of the subsequent sensor data with respect to a prediction of the subsequent sensor data by the statistical model.

It is another object to provide a system for communicating information, comprising: a predictive statistical model, stored in a memory, derived by modeling a stream of sensor data; a communication port configured to communicate with a receiver; and at least one processor, configured to: transmit information defining the predictive statistical model to the receiver; and after communication of the information defining the predictive statistical model to the receiver, communicate information characterizing subsequent sensor data to the receiver, dependent on an error of the subsequent sensor data with respect to a prediction of the subsequent sensor data by the statistical model.

A further object provides a system for synchronizing a state of a transmitter and a receiver, to communicate a stream of sensor data, comprising: a communication port configured to communicate with a receiver; and at least one automated processor, configured to: model the stream of sensor data, and to define parameters of a predictive statistical model; communicate the defined parameters of a predictive statistical model to the receiver; and communicate information characterizing subsequent sensor data to the receiver, comprising a series of statistically normalized differentially encoded subsequent sensor data with respect to a prediction of the series of subsequent sensor data by the predictive statistical model.

The method may further comprise calculating, at the receiver, the subsequent sensor data from the error of the sensor data and the prediction of the sensor data by statistical model.

The method may further comprise acquiring a time series of subsequent sensor data, and communicating from the transmitter to the receiver, information characterizing the time series of subsequent sensor data comprising a time series of errors of subsequent sensor data time-samples with respect to a prediction of the subsequent sensor data time-samples by the predictive statistical model.

The predictive statistical model may be adaptive to the communicated information characterizing subsequent sensor data.

The method may further comprise storing information dependent on the predictive statistical model in a memory of the transmitter and a memory of the receiver.

The method may further comprise determining a sensor data standard error based on a predicted sensor data error standard deviation.

The predictive statistical model may be derived from a machine learning based algorithm developed based on relationships between independent and dependent variables represented in the sensor data.

The predictive statistical model may generate error statistics comprising a mean training error and a standard deviation of the mean training error for a stream of sensor data of the training data set in a training period.

The predictive statistical model may comprise a linear model generated by machine learning.

The predictive statistical model may comprise a plurality of predictive statistical models, each provided for a subset of a range of at least one independent variable of the steam of sensor data.

The method may further comprise computing a predicted stream of sensor data, a predicted stream of sensor data error means, and a predicted stream of sensor data error standard deviations, based on the predictive statistical model.

The method may further comprise communicating the predicted stream of sensor data error means from the transmitter to the receiver. The method may further comprise receiving the predicted stream of sensor data error means at the receiver, and based on the predictive statistical model and the received stream of sensor data error means, reconstructing the stream of sensor data.

The method may further comprise approximately reconstructing a stream of subsequent sensor data based on the received predictive statistical model, at least one control variable, and the errors of stream of subsequent sensor data.

The method may further comprise transmitting a standard error of the prediction of the subsequent sensor data by the predictive statistical model from the transmitter to the receiver, and inferring the prediction of the subsequent sensor data by the predictive statistical model at the receiver from the received standard error of the prediction and the predictive statistical model.

The stream of sensor data may comprise sensor data from a plurality of sensors which are dependent on at least one common control variable, the predictive statistical model being dependent on a correlation of the sensor data from the plurality of sensors, further comprise calculating standard errors of the subsequent sensor data from the plurality of sensors with respect to the predictive statistical model dependent on a correlation of the sensor data, entropy encoding the standard errors based on at least the correlation, and transmitting the entropy encoded standard errors, and a representation of the at least one common control variable from the transmitter to the receiver.

The stream of sensor data comprises engine data. The engine data may comprise timestamped data comprise at least one of engine speed, engine load, coolant temperature, coolant pressure, oil temperature, oil pressure, fuel pressure, and fuel actuator state. The engine data may comprise timestamped data comprise engine speed, engine load percentage, and at least one of coolant temperature, coolant pressure, oil temperature, oil pressure, and fuel pressure. The engine may be a diesel engine, and the modeled stream of sensor data is acquired while the diesel engine is in a steady state within a bounded range of engine speed and engine load.

The predictive statistical model may be a spline model, a neural network, a support vector machine, and/or a Generalized Additive Model (GAM).

Various predictive modeling methods are known, including Group method of data handling; Naive Bayes; k-nearest neighbor algorithm; Majority classifier; Support vector machines; Random forests; Boosted trees; CART (Classification and Regression Trees); Multivariate adaptive regression splines (MARS); Neural Networks and deep neural networks; ACE and AVAS; Ordinary Least Squares; Generalized Linear Models (GLM) (The generalized linear model (GLM) is a flexible family of models that are unified under a single method. Logistic regression is a notable special case of GLM. Other types of GLM include Poisson regression, gamma regression, and multinomial regression); Logistic regression (Logistic regression is a technique in which unknown values of a discrete variable are predicted based on known values of one or more continuous and/or discrete variables. Logistic regression differs from ordinary least squares (OLS) regression in that the dependent variable is binary in nature. This procedure has many applications); Generalized additive models; Robust regression; and Semiparametric regression. See:

Geisser, Seymour (September 2016). Predictive Inference: An Introduction. New York: Chapman & Hall. ISBN 0-412-03471-9.

Finlay, Steven (2014). Predictive Analytics, Data Mining and Big Data. Myths, Misconceptions and Methods (1st ed.). Basingstoke: Palgrave Macmillan. p. 237. ISBN 978-1137379276.

Sheskin, David J. (April 27, 2011). Handbook of Parametric and Nonparametric Statistical Procedures. Boca Raton, FL: CRC Press. p. 109. ISBN 1439858012.

Marascuilo, Leonard A. (December 1977). Nonparametric and distribution-free methods for the social sciences. Brooks/Cole Publishing Co. ISBN 0818502029.

Wilcox, Rand R. (March 18, 2010). Fundamentals of Modern Statistical Methods. New York: Springer. pp. 200–213. ISBN 1441955240.

Steyerberg, Ewout W. (October 21, 2010). Clinical Prediction Models. New York: Springer. p. 313. ISBN 1441926488.

Breiman, Leo (August 1996). “Bagging predictors”. Machine Learning. 24 (2): 123–140. doi:10.1007/bf00058655.

Willey, Gordon R. (1953) “Prehistoric Settlement Patterns in the Virú Valley, Peru”, Bulletin 155. Bureau of American Ethnology

Heidelberg, Kurt, et al. “An Evaluation of the Archaeological Sample Survey Program at the Nevada Test and Training Range”, SRI Technical Report 02-16, 2002

Jeffrey H. Altschul, Lynne Sebastian, and Kurt Heidelberg, “Predictive Modeling in the Military: Similar Goals, Divergent Paths”, Preservation Research Series 1, SRI Foundation, 2004

forteconsultancy.wordpress.com/2010/05/17/wondering-what-lies-ahead-the-power-of-predictive-modeling/

“Hospital Uses Data Analytics and Predictive Modeling To Identify and Allocate Scarce Resources to High-Risk Patients, Leading to Fewer Readmissions”. Agency for Healthcare Research and Quality. 2014-01-29. Retrieved 2014-01-29.

Banerjee, Imon. “Probabilistic Prognostic Estimates of Survival in Metastatic Cancer Patients (PPES-Met) Utilizing Free-Text Clinical Narratives”. Scientific Reports. 8 (10037 (2018)). doi:10.1038/s41598-018-27946-5.

“Implementing Predictive Modeling in R for Algorithmic Trading”. 2016-10-07. Retrieved 2016-11-25.

“Predictive-Model Based Trading Systems, Part 1 - System Trader Success”. System Trader Success. 2013-07-22. Retrieved 2016-11-25.

10061887; 10126309; 10154624; 10168337; 10187899; 6006182; 6064960; 6366884; 6401070; 6553344; 6785652; 7039654; 7144869; 7379890; 7389114; 7401057; 7426499; 7547683; 7561972; 7561973; 7583961; 7653491; 7693683; 7698213; 7702576; 7729864; 7730063; 7774272; 7813981; 7873567; 7873634; 7970640; 8005620; 8126653; 8152750; 8185486; 8401798; 8412461; 8498915; 8515719; 8566070; 8635029; 8694455; 8713025; 8724866; 8731728; 8843356; 8929568; 8992453; 9020866; 9037256; 9075796; 9092391; 9103826; 9204319; 9205064; 9297814; 9428767; 9471884; 9483531; 9534234; 9574209; 9580697; 9619883; 9886545; 9900790; 9903193; 9955488; 9992123; 20010009904; 20010034686; 20020001574; 20020138012; 20020138270; 20030023951; 20030093277; 20040073414; 20040088239; 20040110697; 20040172319; 20040199445; 20040210509; 20040215551; 20040225629; 20050071266; 20050075597; 20050096963; 20050144106; 20050176442; 20050245252; 20050246314; 20050251468; 20060059028; 20060101017; 20060111849; 20060122816; 20060136184; 20060184473; 20060189553; 20060241869; 20070038386; 20070043656; 20070067195; 20070105804; 20070166707; 20070185656; 20070233679; 20080015871; 20080027769; 20080027841; 20080050357; 20080114564; 20080140549; 20080228744; 20080256069; 20080306804; 20080313073; 20080319897; 20090018891; 20090030771; 20090037402; 20090037410; 20090043637; 20090050492; 20090070182; 20090132448; 20090171740; 20090220965; 20090271342; 20090313041; 20100028870; 20100029493; 20100042438; 20100070455; 20100082617; 20100100331; 20100114793; 20100293130; 20110054949; 20110059860;

20110064747; 20110075920; 20110111419; 20110123986; 20110123987; 20110166844; 20110230366; 20110276828; 20110287946; 20120010867; 20120066217; 20120136629; 20120150032; 20120158633; 20120207771; 20120220958; 20120230515; 20120258874; 20120283885; 20120284207; 20120290505; 20120303408; 20120303504; 20130004473; 20130030584; 20130054486; 20130060305; 20130073442; 20130096892; 20130103570; 20130132163; 20130183664; 20130185226; 20130259847; 20130266557; 20130315885; 20140006013; 20140032186; 20140100128; 20140172444; 20140193919; 20140278967; 20140343959; 20150023949; 20150235143; 20150240305; 20150289149; 20150291975; 20150291976; 20150291977; 20150316562; 20150317449; 20150324548; 20150347922; 20160003845; 20160042513; 20160117327; 20160145693; 20160148237; 20160171398; 20160196587; 20160225073; 20160225074; 20160239919; 20160282941; 20160333328; 20160340691; 20170046347; 20170126009; 20170132537; 20170137879; 20170191134; 20170244777; 20170286594; 20170290024; 20170306745; 20170308672; 20170308846; 20180006957; 20180017564; 20180018683; 20180035605; 20180046926; 20180060458; 20180060738; 20180060744; 20180120133; 20180122020; 20180189564; 20180227930; 20180260515; 20180260717; 20180262433; 20180263606; 20180275146; 20180282736; 20180293511; 20180334721; 20180341958; 20180349514; 20190010554; and 20190024497.

In statistics, the generalized linear model (GLM) is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution. The GLM generalizes linear regression by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value. Generalized linear models unify various other statistical models, including linear regression, logistic regression and Poisson regression, and employs an iteratively reweighted least squares method for maximum likelihood estimation of the model parameters. See:

10002367; 10006088; 10009366; 10013701; 10013721; 10018631; 10019727; 10021426; 10023877; 10036074; 10036638; 10037393; 10038697; 10047358; 10058519; 10062121; 10070166; 10070220; 10071151; 10080774; 10092509; 10098569; 10098908; 10100092; 10101340; 10111888; 10113198; 10113200; 10114915; 10117868; 10131949; 10142788; 10147173; 10157509; 10172363; 10175387; 10181010; 5529901; 5641689; 5667541; 5770606; 5915036; 5985889; 6043037; 6121276; 6132974; 6140057; 6200983; 6226393; 6306437; 6411729; 6444870; 6519599; 6566368; 6633857; 6662185; 6684252; 6703231; 6704718; 6879944; 6895083; 6939670; 7020578; 7043287; 7069258; 7117185; 7179797; 7208517; 7228171; 7238799; 7268137; 7306913; 7309598; 7337033; 7346507; 7445896; 7473687; 7482117; 7494783; 7516572; 7550504; 7590516; 7592507; 7593815; 7625699; 7651840; 7662564; 7685084; 7693683; 7695911; 7695916; 7700074; 7702482; 7709460; 7711488; 7727725; 7743009; 7747392; 7751984; 7781168; 7799530; 7807138; 7811794; 7816083; 7820380; 7829282; 7833706; 7840408; 7853456; 7863021; 7888016; 7888461; 7888486; 7890403; 7893041; 7904135; 7910107; 7910303; 7913556; 7915244; 7921069; 7933741; 7947451; 7953676; 7977052; 7987148; 7993833; 7996342; 8010476; 8017317; 8024125; 8027947; 8037043; 8039212; 8071291; 8071302; 8094713; 8103537; 8135548; 8148070; 8153366; 8211638; 8214315; 8216786; 8217078; 8222270; 8227189; 8234150; 8234151; 8236816; 8283440; 8291069; 8299109; 8311849; 8328950; 8346688; 8349327; 8351688; 8364627; 8372625; 8374837; 8383338; 8412465; 8415093; 8434356; 8452621; 8452638; 8455468; 8461849; 8463582; 8465980; 8473249; 8476077; 8489499; 8496934; 8497084; 8501718; 8501719; 8514928; 8515719; 8521294; 8527352; 8530831; 8543428; 8563295; 8566070; 8568995; 8569574; 8600870; 8614060; 8618164; 8626697; 8639618; 8645298; 8647819; 8652776; 8669063; 8682812; 8682876; 8706589; 8712937; 8715704; 8715943; 8718958; 8725456; 8725541; 8731977; 8732534; 8741635; 8741956; 8754805; 8769094; 8787638; 8799202; 8805619; 8811670; 8812362; 8822149; 8824762; 8871901; 8877174; 8889662; 8892409; 8903192; 8903531; 8911958; 8912512; 8956608; 8962680; 8965625; 8975022; 8977421; 8987686; 9011877;

9030565; 9034401; 9036910; 9037256; 9040023; 9053537; 9056115; 9061004; 9061055; 9069352; 9072496; 9074257; 9080212; 9106718; 9116722; 9128991; 9132110; 9186107; 9200324; 9205092; 9207247; 9208209; 9210446; 9211103; 9216010; 9216213; 9226518; 9232217; 9243493; 9275353; 9292550; 9361274; 9370501; 9370509; 9371565; 9374671; 9375412; 9375436; 9389235; 9394345; 9399061; 9402871; 9415029; 9451920; 9468541; 9503467; 9534258; 9536214; 9539223; 9542939; 9555069; 9555251; 9563921; 9579337; 9585868; 9615585; 9625646; 9633401; 9639807; 9639902; 9650678; 9663824; 9668104; 9672474; 9674210; 9675642; 9679378; 9681835; 9683832; 9701721; 9710767; 9717459; 9727616; 9729568; 9734122; 9734290; 9740979; 9746479; 9757388; 9758828; 9760907; 9769619; 9775818; 9777327; 9786012; 9790256; 9791460; 9792741; 9795335; 9801857; 9801920; 9809854; 9811794; 9836577; 9870519; 9871927; 9881339; 9882660; 9886771; 9892420; 9926368; 9926593; 9932637; 9934239; 9938576; 9949659; 9949693; 9951348; 9955190; 9959285; 9961488; 9967714; 9972014; 9974773; 9976182; 9982301; 9983216; 9986527; 9988624; 9990648; 9990649; 9993735; 20020016699; 20020055457; 20020099686; 20020184272; 20030009295; 20030021848; 20030023951; 20030050265; 20030073715; 20030078738; 20030104499; 20030139963; 20030166017; 20030166026; 20030170660; 20030170700; 20030171685; 20030171876; 20030180764; 20030190602; 20030198650; 20030199685; 20030220775; 20040063095; 20040063655; 20040073414; 20040092493; 20040115688; 20040116409; 20040116434; 20040127799; 20040138826; 20040142890; 20040157783; 20040166519; 20040265849; 20050002950; 20050026169; 20050080613; 20050096360; 20050113306; 20050113307; 20050164206; 20050171923; 20050272054; 20050282201; 20050287559; 20060024700; 20060035867; 20060036497; 20060084070; 20060084081; 20060142983; 20060143071; 20060147420; 20060149522; 20060164997; 20060223093; 20060228715; 20060234262; 20060278241; 20060286571; 20060292547; 20070026426; 20070031846; 20070031847; 20070031848; 20070036773; 20070037208; 20070037241; 20070042382; 20070049644; 20070054278; 20070059710;

20070065843; 20070072821; 20070078117; 20070078434; 20070087000; 20070088248; 20070123487; 20070129948; 20070167727; 20070190056; 20070202518; 20070208600; 20070208640; 20070239439; 20070254289; 20070254369; 20070255113; 20070259954; 20070275881; 20080032628; 20080033589; 20080038230; 20080050732; 20080050733; 20080051318; 20080057500; 20080059072; 20080076120; 20080103892; 20080108081; 20080108713; 20080114564; 20080127545; 20080139402; 20080160046; 20080166348; 20080172205; 20080176266; 20080177592; 20080183394; 20080195596; 20080213745; 20080241846; 20080248476; 20080286796; 20080299554; 20080301077; 20080305967; 20080306034; 20080311572; 20080318219; 20080318914; 20090006363; 20090035768; 20090035769; 20090035772; 20090053745; 20090055139; 20090070081; 20090076890; 20090087909; 20090089022; 20090104620; 20090107510; 20090112752; 20090118217; 20090119357; 20090123441; 20090125466; 20090125916; 20090130682; 20090131702; 20090132453; 20090136481; 20090137417; 20090157409; 20090162346; 20090162348; 20090170111; 20090175830; 20090176235; 20090176857; 20090181384; 20090186352; 20090196875; 20090210363; 20090221438; 20090221620; 20090226420; 20090233299; 20090253952; 20090258003; 20090264453; 20090270332; 20090276189; 20090280566; 20090285827; 20090298082; 20090306950; 20090308600; 20090312410; 20090325920; 20100003691; 20100008934; 20100010336; 20100035983; 20100047798; 20100048525; 20100048679; 20100063851; 20100076949; 20100113407; 20100120040; 20100132058; 20100136553; 20100136579; 20100137409; 20100151468; 20100174336; 20100183574; 20100183610; 20100184040; 20100190172; 20100191216; 20100196400; 20100197033; 20100203507; 20100203508; 20100215645; 20100216154; 20100216655; 20100217648; 20100222225; 20100249188; 20100261187; 20100268680; 20100272713; 20100278796; 20100284989; 20100285579; 20100310499; 20100310543; 20100330187; 20110004509; 20110021555; 20110027275; 20110028333; 20110054356; 20110065981; 20110070587;

20110071033; 20110077194; 20110077215; 20110077931; 20110079077; 20110086349; 20110086371; 20110086796; 20110091994; 20110093288; 20110104121; 20110106736; 20110118539; 20110123100; 20110124119; 20110129831; 20110130303; 20110131160; 20110135637; 20110136260; 20110137851; 20110150323; 20110173116; 20110189648; 20110207659; 20110207708; 20110208738; 20110213746; 20110224181; 20110225037; 20110251272; 20110251995; 20110257216; 20110257217; 20110257218; 20110257219; 20110263633; 20110263634; 20110263635; 20110263636; 20110263637; 20110269735; 20110276828; 20110284029; 20110293626; 20110302823; 20110307303; 20110311565; 20110319811; 20120003212; 20120010274; 20120016106; 20120016436; 20120030082; 20120039864; 20120046263; 20120064512; 20120065758; 20120071357; 20120072781; 20120082678; 20120093376; 20120101965; 20120107370; 20120108651; 20120114211; 20120114620; 20120121618; 20120128223; 20120128702; 20120136629; 20120154149; 20120156215; 20120163656; 20120165221; 20120166291; 20120173200; 20120184605; 20120209565; 20120209697; 20120220055; 20120239489; 20120244145; 20120245133; 20120250963; 20120252050; 20120252695; 20120257164; 20120258884; 20120264692; 20120265978; 20120269846; 20120276528; 20120280146; 20120301407; 20120310619; 20120315655; 20120316833; 20120330720; 20130012860; 20130024124; 20130024269; 20130029327; 20130029384; 20130030051; 20130040922; 20130040923; 20130041034; 20130045198; 20130045958; 20130058914; 20130059827; 20130059915; 20130060305; 20130060549; 20130061339; 20130065870; 20130071033; 20130073213; 20130078627; 20130080101; 20130081158; 20130102918; 20130103615; 20130109583; 20130112895; 20130118532; 20130129764; 20130130923; 20130138481; 20130143215; 20130149290; 20130151429; 20130156767; 20130171296; 20130197081; 20130197738; 20130197830; 20130198203; 20130204664; 20130204833; 20130209486; 20130210855; 20130211229; 20130212168; 20130216551; 20130225439; 20130237438; 20130237447; 20130240722;

20130244233; 20130244902; 20130244965; 20130252267; 20130252822; 20130262425; 20130271668; 20130273103; 20130274195; 20130280241; 20130288913; 20130303558; 20130303939; 20130310261; 20130315894; 20130325498; 20130332231; 20130332338; 20130346023; 20130346039; 20130346844; 20140004075; 20140004510; 20140011206; 20140011787; 20140038930; 20140058528; 20140072550; 20140072957; 20140080784; 20140081675; 20140086920; 20140087960; 20140088406; 20140093127; 20140093974; 20140095251; 20140100989; 20140106370; 20140107850; 20140114746; 20140114880; 20140120137; 20140120533; 20140127213; 20140128362; 20140134186; 20140134625; 20140135225; 20140141988; 20140142861; 20140143134; 20140148505; 20140156231; 20140156571; 20140163096; 20140170069; 20140171337; 20140171382; 20140172507; 20140178348; 20140186333; 20140188918; 20140199290; 20140200953; 20140200999; 20140213533; 20140219968; 20140221484; 20140234291; 20140234347; 20140235605; 20140236965; 20140242180; 20140244216; 20140249447; 20140249862; 20140256576; 20140258355; 20140267700; 20140271672; 20140274885; 20140278148; 20140279053; 20140279306; 20140286935; 20140294903; 20140303481; 20140316217; 20140323897; 20140324521; 20140336965; 20140343786; 20140349984; 20140365144; 20140365276; 20140376645; 20140378334; 20150001420; 20150002845; 20150004641; 20150005176; 20150006605; 20150007181; 20150018632; 20150019262; 20150025328; 20150031578; 20150031969; 20150032598; 20150032675; 20150039265; 20150051896; 20150051949; 20150056212; 20150064194; 20150064195; 20150064670; 20150066738; 20150072434; 20150072879; 20150073306; 20150078460; 20150088783; 20150089399; 20150100407; 20150100408; 20150100409; 20150100410; 20150100411; 20150100412; 20150111775; 20150112874; 20150119759; 20150120758; 20150142331; 20150152176; 20150167062; 20150169840; 20150178756; 20150190367; 20150190436; 20150191787; 20150205756; 20150209586; 20150213192; 20150215127; 20150216164; 20150216922; 20150220487;

20150228031; 20150228076; 20150231191; 20150232944; 20150240304; 20150240314; 20150250816; 20150259744; 20150262511; 20150272464; 20150287143; 20150292010; 20150292016; 20150299798; 20150302529; 20150306160; 20150307614; 20150320707; 20150320708; 20150328174; 20150332013; 20150337373; 20150341379; 20150348095; 20150356458; 20150359781; 20150361494; 20150366830; 20150377909; 20150378807; 20150379428; 20150379429; 20150379430; 20160010162; 20160012334; 20160017037; 20160017426; 20160024575; 20160029643; 20160029945; 20160032388; 20160034640; 20160034664; 20160038538; 20160040184; 20160040236; 20160042009; 20160042197; 20160045466; 20160046991; 20160048925; 20160053322; 20160058717; 20160063144; 20160068890; 20160068916; 20160075665; 20160078361; 20160097082; 20160105801; 20160108473; 20160108476; 20160110657; 20160110812; 20160122396; 20160124933; 20160125292; 20160138105; 20160139122; 20160147013; 20160152538; 20160163132; 20160168639; 20160171618; 20160171619; 20160173122; 20160175321; 20160198657; 20160202239; 20160203279; 20160203316; 20160222100; 20160222450; 20160224724; 20160224869; 20160228056; 20160228392; 20160237487; 20160243190; 20160243215; 20160244836; 20160244837; 20160244840; 20160249152; 20160250228; 20160251720; 20160253324; 20160253330; 20160259883; 20160265055; 20160271144; 20160281105; 20160281164; 20160282941; 20160295371; 20160303111; 20160303172; 20160306075; 20160307138; 20160310442; 20160319352; 20160344738; 20160352768; 20160355886; 20160359683; 20160371782; 20160378942; 20170004409; 20170006135; 20170007574; 20170009295; 20170014032; 20170014108; 20170016896; 20170017904; 20170022563; 20170022564; 20170027940; 20170028006; 20170029888; 20170029889; 20170032100; 20170035011; 20170037470; 20170046499; 20170051019; 20170051359; 20170052945; 20170056468; 20170061073; 20170067121; 20170068795; 20170071884; 20170073756; 20170074878; 20170076303; 20170088900; 20170091673; 20170097347; 20170098240;

20170098257; 20170098278; 20170099836; 20170100446; 20170103190; 20170107583; 20170108502; 20170112792; 20170116624; 20170116653; 20170117064; 20170119662; 20170124520; 20170124528; 20170127110; 20170127180; 20170135647; 20170140122; 20170140424; 20170145503; 20170151217; 20170156344; 20170157249; 20170159045; 20170159138; 20170168070; 20170177813; 20170180798; 20170193647; 20170196481; 20170199845; 20170214799; 20170219451; 20170224268; 20170226164; 20170228810; 20170231221; 20170233809; 20170233815; 20170235894; 20170236060; 20170238850; 20170238879; 20170242972; 20170246963; 20170247673; 20170255888; 20170255945; 20170259178; 20170261645; 20170262580; 20170265044; 20170268066; 20170270580; 20170280717; 20170281747; 20170286594; 20170286608; 20170286838; 20170292159; 20170298126; 20170300814; 20170300824; 20170301017; 20170304248; 20170310697; 20170311895; 20170312289; 20170312315; 20170316150; 20170322928; 20170344554; 20170344555; 20170344556; 20170344954; 20170347242; 20170350705; 20170351689; 20170351806; 20170351811; 20170353825; 20170353826; 20170353827; 20170353941; 20170363738; 20170364596; 20170364817; 20170369534; 20170374521; 20180000102; 20180003722; 20180005149; 20180010136; 20180010185; 20180010197; 20180010198; 20180011110; 20180014771; 20180017545; 20180017564; 20180017570; 20180020951; 20180021279; 20180031589; 20180032876; 20180032938; 20180033088; 20180038994; 20180049636; 20180051344; 20180060513; 20180062941; 20180064666; 20180067010; 20180067118; 20180071285; 20180075357; 20180077146; 20180078605; 20180080081; 20180085168; 20180085355; 20180087098; 20180089389; 20180093418; 20180093419; 20180094317; 20180095450; 20180108431; 20180111051; 20180114128; 20180116987; 20180120133; 20180122020; 20180128824; 20180132725; 20180143986; 20180148776; 20180157758; 20180160982; 20180171407; 20180182181; 20180185519; 20180191867; 20180192936; 20180193652; 20180201948; 20180206489; 20180207248; 20180214404;

20180216099; 20180216100; 20180216101; 20180216132; 20180216197; 20180217141; 20180217143; 20180218117; 20180225585; 20180232421; 20180232434; 20180232661; 20180232700; 20180232702; 20180232904; 20180235549; 20180236027; 20180237825; 20180239829; 20180240535; 20180245154; 20180251819; 20180251842; 20180254041; 20180260717; 20180263962; 20180275629; 20180276325; 20180276497; 20180276498; 20180276570; 20180277146; 20180277250; 20180285765; 20180285900; 20180291398; 20180291459; 20180291474; 20180292384; 20180292412; 20180293462; 20180293501; 20180293759; 20180300333; 20180300639; 20180303354; 20180303906; 20180305762; 20180312923; 20180312926; 20180314964; 20180315507; 20180322203; 20180323882; 20180327740; 20180327806; 20180327844; 20180336534; 20180340231; 20180344841; 20180353138; 20180357361; 20180357362; 20180357529; 20180357565; 20180357726; 20180358118; 20180358125; 20180358128; 20180358132; 20180359608; 20180360892; 20180365521; 20180369238; 20180369696; 20180371553; 20190000750; 20190001219; 20190004996; 20190005586; 20190010548; 20190015035; 20190017117; 20190017123; 20190024174; 20190032136; 20190033078; 20190034473; 20190034474; 20190036779; 20190036780; and 20190036816.

Ordinary linear regression predicts the expected value of a given unknown quantity (the response variable, a random variable) as a linear combination of a set of observed values (predictors). This implies that a constant change in a predictor leads to a constant change in the response variable (*i.e.*, a linear-response model). This is appropriate when the response variable has a normal distribution (intuitively, when a response variable can vary essentially indefinitely in either direction with no fixed “zero value”, or more generally for any quantity that only varies by a relatively small amount, *e.g.*, human heights). However, these assumptions are inappropriate for some types of response variables. For example, in cases where the response variable is expected to be always positive and varying over a wide range, constant input changes lead to geometrically varying, rather than constantly varying, output changes.

In a GLM, each outcome Y of the dependent variables is assumed to be generated from a particular distribution in the exponential family, a large range of probability distributions that includes the normal, binomial, Poisson and gamma distributions, among others.

The GLM consists of three elements: A probability distribution from the exponential family; a linear predictor $\eta = X\beta$; and a link function g such that $E(Y) = \mu = g{-}1(\eta)$. The linear predictor is the quantity which incorporates the information about the independent variables into the model. The symbol η (Greek "eta") denotes a linear predictor. It is related to the expected value of the data through the link function. η is expressed as linear combinations (thus, "linear") of unknown parameters β. The coefficients of the linear combination are represented as the matrix of independent variables X. η can thus be expressed as The link function provides the relationship between the linear predictor and the mean of the distribution function. There are many commonly used link functions, and their choice is informed by several considerations. There is always a well-defined canonical link function which is derived from the exponential of the response's density function. However, in some cases it makes sense to try to match the domain of the link function to the range of the distribution function's mean, or use a non-canonical link function for algorithmic purposes, for example Bayesian probit regression. For the most common distributions, the mean is one of the parameters in the standard form of the distribution's density function, and then is the function as defined above that maps the density function into its canonical form. A simple, very important example of a generalized linear model (also an example of a general linear model) is linear regression. In linear regression, the use of the least-squares estimator is justified by the Gauss–Markov theorem, which does not assume that the distribution is normal.

The standard GLM assumes that the observations are uncorrelated. Extensions have been developed to allow for correlation between observations, as occurs for example in longitudinal studies and clustered designs. Generalized estimating equations (GEEs) allow for the correlation between observations without the use of an explicit probability model for the origin of the correlations, so there is no explicit likelihood. They are suitable when the random effects and their variances are not of inherent interest, as they allow for the correlation without explaining its origin. The focus is on estimating the average response over the population (“population-averaged” effects) rather than the regression parameters that would enable prediction of the effect of changing one or more components of X on a given individual. GEEs are usually used in conjunction with Huber-White standard errors. Generalized linear mixed models (GLMMs) are an extension to GLMs that includes random effects in the linear predictor, giving an explicit probability model that explains the origin of the correlations. The resulting “subject-specific” parameter estimates are suitable when the focus is on estimating the effect of changing one or more components of X on a given individual. GLMMs are also referred to as multilevel models and as mixed model. In general, fitting GLMMs is more computationally complex and intensive than fitting GEEs.

In statistics, a generalized additive model (GAM) is a generalized linear model in which the linear predictor depends linearly on unknown smooth functions of some predictor variables, and interest focuses on inference about these smooth functions. GAMs were originally developed by Trevor Hastie and Robert Tibshirani to blend properties of generalized linear models with additive models.

The model relates a univariate response variable, to some predictor variables. An exponential family distribution is specified for (for example normal, binomial or Poisson distributions) along with a link function g (for example the identity or log functions) relating the expected value of univariate response variable to the predictor variables.

The functions may have a specified parametric form (for example a polynomial, or an un-penalized regression spline of a variable) or may be specified non-parametrically, or semi-parametrically, simply as ‘smooth functions’, to be estimated by non-parametric means. So a typical GAM might use a scatterplot smoothing function, such as a locally weighted mean. This flexibility to allow non-parametric fits with relaxed assumptions on the actual relationship between response and predictor, provides the potential for better fits to data than purely parametric models, but arguably with some loss of interpretability.

Any multivariate function can be represented as sums and compositions of univariate functions. Unfortunately, though the Kolmogorov–Arnold representation theorem asserts the existence of a function of this form, it gives no mechanism whereby one could be constructed. Certain constructive proofs exist, but they tend to require highly complicated (*i.e.*, fractal) functions, and thus are not suitable for modeling approaches. It is not clear that any step-wise (*i.e.*, backfitting algorithm) approach could even approximate a solution. Therefore, the Generalized Additive Model drops the outer sum, and demands instead that the function belong to a simpler class.

The original GAM fitting method estimated the smooth components of the model using non-parametric smoothers (for example smoothing splines or local linear regression smoothers) via the backfitting algorithm. Backfitting works by iterative smoothing of partial residuals and provides a very general modular estimation method capable of using a wide variety of smoothing methods to estimate the terms. Many modern implementations of GAMs and their extensions are built around the reduced rank smoothing approach, because it allows well founded estimation of the smoothness of the component smooths at comparatively modest computational cost, and also facilitates implementation of a number of model extensions in a way that is more difficult with other methods. At its simplest the idea is to replace the unknown smooth functions in the model with basis expansions. Smoothing bias complicates interval estimation for these models, and the simplest approach turns out to involve a Bayesian approach. Understanding this Bayesian view of smoothing also helps to understand the REML and full Bayes approaches to smoothing parameter estimation. At some level smoothing penalties are imposed.

Overfitting can be a problem with GAMs, especially if there is un-modelled residual auto-correlation or un-modelled overdispersion. Cross-validation can be used to detect and/or reduce overfitting problems with GAMs (or other statistical methods), and software often allows the level of penalization to be increased to force smoother fits. Estimating very large numbers of smoothing parameters is also likely to be statistically challenging, and there are known tendencies for prediction error criteria (GCV, AIC etc.) to occasionally undersmooth substantially, particularly at moderate sample sizes, with REML being somewhat less problematic in this regard. Where appropriate, simpler models such as GLMs may be preferable to GAMs unless GAMs improve predictive ability substantially (in validation sets) for the application in question.

Augustin, N.H.; Sauleau, E-A; Wood, S.N. (2012). “On quantile quantile plots for generalized linear models”. Computational Statistics and Data Analysis. 56: 2404–2409. doi:10.1016/j.csda.2012.01.026.

Brian Junker (March 22, 2010). “Additive models and cross-validation”.

Chambers, J.M.; Hastie, T. (1993). Statistical Models in S. Chapman and Hall.

Dobson, A.J.; Barnett, A.G. (2008). Introduction to Generalized Linear Models (3rd ed.). Boca Raton, FL: Chapman and Hall/CRC. ISBN 1-58488-165-8.

Fahrmeier, L.; Lang, S. (2001). “Bayesian Inference for Generalized Additive Mixed Models based on Markov Random Field Priors”. Journal of the Royal Statistical Society, Series C. 50: 201–220.

Greven, Sonja; Kneib, Thomas (2010). “On the behaviour of marginal and conditional AIC in linear mixed models”. Biometrika. 97: 773–789. doi:10.1093/biomet/asq042.

Gu, C.; Wahba, G. (1991). “Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method”. SIAM Journal on Scientific and Statistical Computing. 12. pp. 383–398.

Gu, Chong (2013). Smoothing Spline ANOVA Models (2nd ed.). Springer.

Hardin, James; Hilbe, Joseph (2003). Generalized Estimating Equations. London: Chapman and Hall/CRC. ISBN 1-58488-307-3.

Hardin, James; Hilbe, Joseph (2007). Generalized Linear Models and Extensions (2nd ed.). College Station: Stata Press. ISBN 1-59718-014-9.

Hastie, T. J.; Tibshirani, R. J. (1990). Generalized Additive Models. Chapman & Hall/CRC. ISBN 978-0-412-34390-2.

Kim, Y.J.; Gu, C. (2004). “Smoothing spline Gaussian regression: more scalable computation via efficient approximation”. Journal of the Royal Statistical Society, Series B. 66. pp. 337–356.

Madsen, Henrik; Thyregod, Poul (2011). Introduction to General and Generalized Linear Models. Chapman & Hall/CRC. ISBN 978-1-4200-9155-7.

Marra, G.; Wood, S.N. (2011). “Practical Variable Selection for Generalized Additive Models”. Computational Statistics and Data Analysis. 55: 2372–2387. doi:10.1016/j.csda.2011.02.004.

Marra, G.; Wood, S.N. (2012). “Coverage properties of confidence intervals for generalized additive model components”. Scandinavian Journal of Statistics. 39: 53–74. doi:10.1111/j.1467-9469.2011.00760.x.

Mayr, A.; Fenske, N.; Hofner, B.; Kneib, T.; Schmid, M. (2012). “Generalized additive models for location, scale and shape for high dimensional data - a flexible approach based on boosting”. Journal of the Royal Statistical Society, Series C. 61: 403–427. doi:10.1111/j.1467-9876.2011.01033.x.

McCullagh, Peter; Nelder, John (1989). Generalized Linear Models, Second Edition. Boca Raton: Chapman and Hall/CRC. ISBN 0-412-31760-5.

Nelder, John; Wedderburn, Robert (1972). “Generalized Linear Models”. Journal of the Royal Statistical Society. Series A (General). Blackwell Publishing. 135 (3): 370–384. doi:10.2307/2344614. JSTOR 2344614.

Nychka, D. (1988). “Bayesian confidence intervals for smoothing splines”. Journal of the American Statistical Association. 83. pp. 1134--1143.

Reiss, P.T.; Ogden, T.R. (2009). “Smoothing parameter selection for a class of semiparametric linear models”. Journal of the Royal Statistical Society, Series B. 71: 505–523. doi:10.1111/j.1467-9868.2008.00695.x.

Rigby, R.A.; Stasinopoulos, D.M. (2005). “Generalized additive models for location, scale and shape (with discussion)”. Journal of the Royal Statistical Society, Series C. 54: 507–554. doi:10.1111/j.1467-9876.2005.00510.x.

Rue, H.; Martino, Sara; Chopin, Nicolas (2009). “Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations (with discussion)”. Journal of the Royal Statistical Society, Series B. 71: 319–392. doi:10.1111/j.1467-9868.2008.00700.x.

Ruppert, D.; Wand, M.P.; Carroll, R.J. (2003). Semiparametric Regression. Cambridge University Press.

Schmid, M.; Hothorn, T. (2008). “Boosting additive models using component-wise P-splines”. Computational Statistics and Data Analysis. 53: 298–311. doi:10.1016/j.csda.2008.09.009.

Senn, Stephen (2003). “A conversation with John Nelder”. Statistical Science. 18 (1): 118–131. doi:10.1214/ss/1056397489.

Silverman, B.W. (1985). “Some Aspects of the Spline Smoothing Approach to Non-Parametric Regression Curve Fitting (with discussion)”. Journal of the Royal Statistical Society, Series B. 47. pp. 1–53.

Umlauf, Nikolaus; Adler, Daniel; Kneib, Thomas; Lang, Stefan; Zeileis, Achim. “Structured Additive Regression Models: An R Interface to BayesX”. Journal of Statistical Software. 63 (21): 1–46.

Wahba, G. (1983). “Bayesian Confidence Intervals for the Cross Validated Smoothing Spline”. Journal of the Royal Statistical Society, Series B. 45. pp. 133–150.

Wahba, Grace. Spline Models for Observational Data. SIAM Rev., 33(3), 502–502 (1991).

Wood, S. N. (2000). “Modelling and smoothing parameter estimation with multiple quadratic penalties”. Journal of the Royal Statistical Society. Series B. 62 (2): 413–428. doi:10.1111/1467-9868.00240.

Wood, S. N. (2017). Generalized Additive Models: An Introduction with R (2nd ed). Chapman & Hall/CRC. ISBN 978-1-58488-474-3.

Wood, S. N.; Pya, N.; Saefken, B. (2016). “Smoothing parameter and model selection for general smooth models (with discussion)”. Journal of the American Statistical Association. 111: 1548–1575. doi:10.1080/01621459.2016.1180986.

Wood, S.N. (2011). “Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models”. Journal of the Royal Statistical Society, Series B. 73: 3–36.

Wood, Simon (2006). Generalized Additive Models: An Introduction with R. Chapman & Hall/CRC. ISBN 1-58488-474-6.

Wood, Simon N. (2008). “Fast stable direct fitting and smoothness selection for generalized additive models”. Journal of the Royal Statistical Society, Series B. 70 (3): 495–518. arXiv:0709.3906. doi:10.1111/j.1467-9868.2007.00646.x.

Yee, Thomas (2015). Vector generalized linear and additive models. Springer. ISBN 978-1-4939-2817-0.

Zeger, Scott L.; Liang, Kung-Yee; Albert, Paul S. (1988). “Models for Longitudinal Data: A Generalized Estimating Equation Approach”. Biometrics. International Biometric Society. 44 (4): 1049–1060. doi:10.2307/2531734. JSTOR 2531734. PMID 3233245.

The stream of sensor data may comprise temporally averaged sensor data for a series of timestamps.

Communications between the transmitter to the receiver may be bandwidth constrained.

The transmitter and receiver may be asymmetric, wherein the transmitter is a data source and the receiver is a data sink, wherein the receiver is configured to receive communications from a plurality of transmitters.

The information characterizing subsequent sensor data may comprise the error of subsequent sensor data with respect to the prediction of the subsequent sensor data by the predictive statistical model comprises a standardized training error mean, standardized by subtracting a training error mean from an instantaneous error between subsequent sensor data and predicted subsequent sensor data, and dividing this difference by a training error standard deviation for a respective sensor, to produce a z-score of a prediction error.

The error of the subsequent sensor data with respect to the prediction of the subsequent sensor data may be statistically normalized and quantized with respect to units of standard deviation away from a predicted mean of the subsequent sensor data.

The error of the subsequent sensor data with respect to the prediction of the subsequent sensor data may be quantized in uneven steps with respect to units of standard deviation away from a predicted mean of the subsequent sensor data.

The error of the subsequent sensor data with respect to the prediction of the subsequent sensor data may be represented with higher resolution for smaller deviation away from a predicted mean of the subsequent sensor data than for higher deviation from the predicted mean of the subsequent sensor data.

The information defining the predictive statistical model communicated from the transmitter to the receiver may be encrypted.

The communicating of information characterizing subsequent sensor data may comprise communicating encrypted information representing independent variables and unencrypted information representing dependent variables.

The at least one processor may be further configured to calculate, the subsequent sensor data from the error of the sensor data and the prediction of the sensor data by statistical model.

The at least one processor may be further configured to acquire a time series of subsequent sensor data, and to communicate to the receiver information characterizing the time series of subsequent sensor data comprise a time series of errors of subsequent sensor data time samples with respect to a prediction of the subsequent sensor data time samples by the predictive statistical model.

The at least one processor may be configured to generate error statistics comprise a mean training error and a standard deviation of the mean training error for a stream of sensor data of the training data set in a training period based on the predictive statistical model.

The at least one processor may be configured to compute a predicted stream of sensor data, a predicted stream of sensor data error means, and a predicted stream of sensor data error standard deviations, based on the predictive statistical model.

The at least one processor may be configured to communicate the predicted stream of sensor data error means to the receiver.

The receiver may be configured to receive the predicted stream of sensor data error means, and based on the predictive statistical model and the received stream of sensor data error means, reconstruct the stream of sensor data.

The receiver may be configured to approximately reconstruct a stream of subsequent sensor data based on the received predictive statistical model, at least one control variable, and the errors of stream of subsequent sensor data.

The at least one processor may be configured to transmit a standard error of the prediction of the subsequent sensor data by the predictive statistical model to the receiver.

The receiver may be configured to infer the prediction of the subsequent sensor data by the predictive statistical model from the received standard error of the prediction and the predictive statistical model.

In accordance with some embodiments, the process of generating model and standard errors, and communicating data based on the generated model and training error statistics (error mean and standard deviation) is as follows.

An edge device collects machine data, such as an n-dimensional engine data time series that may include, but is not limited to, timestamps (ts) and the following engine parameters: engine speed (rpm), engine load percentage (load), coolant temperature (coolant temperature), coolant pressure (coolant pressure), oil temperature (oil temperature), oil pressure (oil pressure), fuel pressure (fuel pressure), and fuel actuator percentage (fuel actuator percentage). The edge device can be a computing node at the “edge” of an enterprise network, metropolitan network, or other network, in accordance with the geographic distribution of computing nodes in a network of, for example, IoT devices. In this aspect, edge computing is a distributed computing paradigm in which computation is largely or completely performed on distributed device nodes as opposed to primarily taking place in a centralized cloud environment.

For example, in accordance with some implementation of the presently disclosed technology, an edge computing device is installed on a vessel and interfaces with the electronic control units/modules (ECUs/ECMs) of all the diesel engines of the vessel. The edge computing device collects engine sensor data as a time series (e.g., all engines' RPMs, load percentages, fuel rates, oil pressures, oil temperatures, coolant pressures, coolant temperatures, air intake temperatures, bearing temperatures, cylinder temperatures, or the like), and collects vessel speed and location data from an internal GPS/DGPS of the edge device and/or the vessel's GPS/DGPS. The edge device can also interface and collect data from onboard PLC and other devices, systems, or assets such as generators, z-drives, tanks, or the like. Illustratively, the edge device collects the sensor data at an approximate rate of sixty samples per minute and aligns the data to every second's time-stamp (e.g., 12:00:00, 12:00:01, 12:00:02, ...) using its own clock that is synchronized via NTP service. For ships, this data is typically transmitted to shore office through satellite connectivity; and for vessels that operate near shore (e.g. inland tugboats) cellular data transmission is another option.

In an example vessel's edge device installation that has 1000 sensor data points, each day the edge device can collect, store and send 24*60*60*1000*4 = 345.6 MB of data at one second resolution (based on a configuration where each sensor data point is 4 bytes or 32 bits in size)! Even if the edge device sends minute's average data (i.e., average or arithmetic mean of every minute's data instead of every second's data), it will transmit 24*60*1000*4 = 5.76 MB a day over a low bandwidth connection (e.g., satellite or cellular), which can still strain low bandwidth network resources – especially when there are multiple vessels transmitting their respective data at the same time.

In various embodiments, the edge devices can reside on vessels, automobiles, aerial vehicles (e.g., planes, drones, etc.), Internet of Things (IoT) devices, or other mobile devices to collect data locally without transmitting the all of the collected data in explicit form to one or more servers, cloud storage, or other remote systems or devices. Referring back to machine data example above, in a variance analysis of diesel engine data, most of the engine parameters, including coolant temperature, are found to have strong correlation with engine RPM and engine load percentage in a bounded range of engine speed, when engine is in steady state, *i.e.*, RPM and engine load is stable. Inside that bounded region of engine RPM (*e.g.*, higher than idle engine RPM), there exists a function $f1$ such that:

coolant temperature = f1(rpm, load)

$f1 : \mathbb{R}^n \mapsto \mathbb{R}^m$. In this case n equals two (rpm and load) and m equals one (coolant temperature)

In other words, $f1$ is a map that allows for prediction of a single dependent variable from two independent variables. Similarly,

coolant pressure = f2(rpm, load) oil temperature = f3(rpm, load)

oil pressure = f4(rpm, load)

fuel pressure = f5(rpm, load)

fuel actuator percentage = f6(rpm, load) fuel rate = f7(rpm, load)

intake temp = f8(rpm, load)

Grouping these maps into one map leads to a multi-dimensional map (i.e., the model) such that f: $\mathbb{R}^n \mapsto \mathbb{R}^m$

where n equals two (rpm, load) and m equals eight (coolant temperature, coolant pressure, oil temperature, oil pressure, fuel pressure, fuel actuator percentage, fuel rate and intake temp) in this case. Critically, many maps are grouped into a single map with the same input variables, enabling potentially many correlated variables (*i.e.*, a tensor of variables) to be predicted within a bounded range. Note that the specific independent variables need not be engine RPM and engine load and need not be limited to two variables. For example, engine operating hours could be added as an independent variable in the map to account for engine degradation with operating time.

In order to create an engine model, a training time period is selected in which the engine had no apparent operational issues. A machine learning-based method is used to generate the engine models on the edge device or in the cloud. For example, a modeling technique is selected that offers low model bias (*e.g.*, spline, neural network or support vector machines (SVM), and/or a Generalized Additive Model (GAM)).

In some embodiments, the programming language ‘R’ is used as an environment for statistical computing and graphics and GAM for creating a low bias model. Error statistics and/or the *z*-scores of the predicted errors are used to further minimize prediction errors. The engine’s operating ranges can be divided into multiple distinct ranges and multiple multi-dimensional models can be built to improve model accuracy.

The same set of training data that was used to build the model (or other applicable training data) is then passed as an input set to the model in order to create a predicted sensor value(s) time series. By subtracting the predicted sensor values from the measured sensor values, an error time series for all the dependent sensor values is created for the training data set. The error statistics, such as mean and standard deviations of the training period error series, are computed and saved as the training period error statistics.

In the event that the data does not comply with presumptions, such as normal distribution, a normalization process may be included. In other cases, alternate statistical techniques may be employed, so long as they are synchronized at transmitter and receiver.

Once the model is deployed to the edge device and the system is operational, the dependent and independent sensor values can be measured in near real-time, and average data (e.g., per minute) may be computed. The expected value for dependent engine sensors can be predicted by passing the independent sensor values to the engine model. The error (*i.e.*, the difference) between the measured value of a dependent variable and its predicted value can then be computed. These errors are standardized by subtracting the training error mean from the instantaneous error and dividing this difference by the training error standard deviations for a given sensor, which is essentially a *z*-score of prediction errors. These *z*-scores of prediction error or standardized prediction error can be sent to a remote computer instead of the actual raw data as measured using a bit description table as described later.

Suppose Y is a set of measured values of a dependent sensor variable, at time-stamps T, where

$T = t_0, t_1, t_2, t_3, t_4, t_5, \ldots$

$Y = y_0, y_1, y_2, y_3, y_4, y_5, \ldots$

X0 and X1 are two independent variables whose values are measured at the same time stamps are

$X0 = x0_0, x0_1, x0_2, x0_3, x0_4, x0_5, \ldots$

$X1 = x1_0, x1_1, x1_2, x1_3, x1_4, x1_5, \ldots$

and a machine learning based model exists $\hat{Y} = f(X0, X1)$

such that the values of Y can be predicted at the same time-stamps by $\hat{Y}$ where

$\hat{Y} = \hat{y}_0, \hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4, \hat{y}_5, \ldots$

such that, $\hat{y}_i = f(x0_i, x1_i)$

suppose the training mean error for sensor Y is μ_Y,

and training error's standard deviation for sensor Y is σ_Y

so the computed standard error series or z-scores of prediction errors will be

$\varepsilon Y = \varepsilon y_1, \varepsilon y_2, \varepsilon y_3, \varepsilon y_4, \varepsilon y_5, \ldots,$

where $\varepsilon y_i = ((y_i - \hat{y}_i) - \mu_Y)/\sigma_Y$

The transmitter (*e.g.*, edge device or sending computer) sends these standard errors along with independent variables X0, X1 and time-stamp data series T. Once these data are received, the receiver computes the predicted sensor value $\hat{y}_i$ at time t_i

$\hat{y}_i = f(x0_i, x1_i)$

where f is the identical machine learning model on both sending and receiving sides. The receiving side can recover a given sensor's value provided that the receiver has the identical machine learning based model as the sender and the training error statistics:

$y_i = \hat{y}_i + \mu_Y + y_i \times \sigma_Y$

By introducing non-linear loss into the compression algorithm, the compression ratios can be greatly increased. As an example, consider the following buckets of standard errors, assigning unique standard error states to unique bit patterns, *e.g.*:

Id	std. err.	Bits
1	0 <= std. err <1	00
2	0 > std. err > -1	01
3	1 <= std. err < 2	1000
4	-1 >= std. err > -2	1100
5	2 <= std. err < 3	1001
6	-2 >= std. err > -3	1101
7	std. err >= 3	1010
8	std. err <= -3	1110
9	Error	1011
10	Null	1111

Four bits represent the value of the standard error when the standard error is outside of the -1 to +1 range and two bits represent the value when standard error is within the -1 to +1 range. The receiver side algorithm can check if the most significant bit (*i.e.*, the left most bit) is zero, thus identifying that the error will be within ±1 and be represented by two bits, otherwise the error will be greater than ±1 and represented by four bits. The second bit determines the polarity of the error quantity (positive or negative standard error) etc.

Using a typical diesel engine as an example, assume that a machine dataset containing 10 sensors must be transmitted. Assume that two of the sensors are independent and eight are dependent sensor variables. Given enough data, a machine learning based model can be generated such that the 8 dependent sensors values can be predicted from an input consisting of 2 independent variables.

Table 1 represents the output of the machine learning based model showing predictions of fuel pressure, fuel actuator percentage, oil temperature, oil pressure, coolant temperature, coolant pressure, fuel rate and intake temperature for an engine speed of 1454 RPM and various engine load percentages.

Table 1: sample engine data averaged every minute

time stamp	engine speed	engine percent load	fuel pressure	fuel actuator pct	oil temperature
2017-05-30 20:16:00	1454	56.93	737.00	38.39	365.34
2017-05-30 20:17:00	1454	56.77	737.00	38.38	365.34
2017-05-30 20:18:00	1454	56.37	737.00	38.34	365.34
2017-05-30 20:19:00	1454	56.97	737.00	38.49	365.34
2017-05-30 20:20:00	1454	56.83	737.00	38.37	365.34
2017-05-30 20:21:00	1454	56.71	737.00	38.32	365.34
2017-05-30 20:22:00	1454	56.40	737.00	38.37	365.34
2017-05-30 20:23:00	1454	56.70	737.00	38.37	365.34
2017-05-30 20:24:00	1454	56.92	737.00	38.40	365.34
2017-05-30 20:25:00	1454	56.44	737.00	38.35	365.34
2017-05-30 20:26:00	1454	56.43	737.00	38.34	365.34

time stamp	oil pressure	coolant temp	coolant pressure	fuel rate	intake temp
2017-05-30 20:16:00	605.00	129.00	45.80	346.43	83.00
2017-05-30 20:17:00	605.00	129.00	44.54	346.33	83.00
2017-05-30 20:18:00	605.00	129.00	45.48	344.84	83.00
2017-05-30 20:19:00	605.00	129.00	45.37	348.59	83.50
2017-05-30 20:20:00	605.00	129.00	45.17	345.73	83.36
2017-05-30 20:21:00	605.00	129.00	45.69	345.40	83.67
2017-05-30 20:22:00	605.00	129.00	45.52	346.60	84.00
2017-05-30 20:23:00	605.00	129.00	46.31	346.22	83.92
2017-05-30 20:24:00	605.00	129.00	46.19	345.56	83.37
2017-05-30 20:25:00	605.00	129.00	46.31	345.92	83.29
2017-05-30 20:26:00	605.00	129.00	45.59	346.48	83.09

Once both sides compute the model, the 2 independent sensor variables, time stamp, and the standard error bucket for each sensor are sent, leading to a total data size of

$$1\, ts * 32\, bits + 2\, Ind.sensors * 32\, bits + N\, Dep.sensors * 4\, bit$$

in the worst case (*i.e.*, when standard error is outside ± 1 range) or a saving of 4 × 8 - 4 = 28 bits per sensor. For 8 sensors, the savings will be 28×8 bits or 28 bytes for each time stamp.

Considering each data row consists of 10 sensors values and a timestamp, the raw size of each machine data row is

$$1\ ts * 32\ bits +\ 2\ Ind.sensors * 32\ bits + 8\ Dep.sensors * 32\ bits = 352\ bits = 44\ bytes$$

which compresses to the following:

$$1\ ts * 32\ bits +\ 2\ Ind.sensors * 32\ bits + 8\ Dep.sensors * 4\ bit = 128\ bits = 16\ bytes$$

or a compression ratio of

$$100 \times (1 - 16/44) = 63.63\%.$$

In the best case, which is also the average case (*i.e.*, when the standard errors are inside the range of ± 1), the standard errors are represented using 2 bits per sensor variable.

So the compressed data size is

$$1\ ts * 32\ bits +\ 2\ Ind.sensors * 32\ bits + 8\ Dep.sensors * 2\ bit = 112\ bits = 14\ bytes$$

or a compression ratio of

$$100 \times (1 - 14/44) = 68.18\%.$$

In general, the compression ratio can be calculated for a machine with m sensors, where n sensors are independent variables, k sensors are dependent variables such that m = n + k, and r sensors are overhead (timestamp, position data, etc.). Assuming all data are 4 bytes. The data size of each row is

$$(m + r) \times 4\ bytes,$$

whereas in the present scheme, the data row size is

$$(n + k/8 + r) \times 4\ bytes$$

producing a compression ratio of

$$100 \times (1 - (n + k/8 + r) \times 4/(m + r) \times 4)$$

for m = 20, n = 2 and k = 18, and r = 1, the above scheme provides worst case compression ratio of

$$100 \times (1 - (2 \times 4 + 18/2 + 4)/(21 \times 4)) = 75.0\%$$

for m = 20, n = 2 and k = 18, and r = 1, the above scheme provides best case and average case compression ratio of

$$100 \times (1 - (2 \times 4 + 18/4 + 4)/(21 \times 4)) = 80.36\%$$

Similarly, for m = 40, n = 2 and k = 38, and r = 1, the above scheme provides best case and average case compression ratio of

$$100 \times (1 - (2 \times 4 + 38/4 + 4)/(41 \times 4)) = 86.89\%$$

Many bucketing schemes of standard errors can be created. For example, ± 1 standard error range may be merged to one state:

Id	**std. err.**	**bits**
1	-1 >= std.err <= 1	0
2	1 <std err <= 2	1000
3	-1 >std err >= -2	1100
4	2 <std err <= 3	1001
5	-2 >std. err >=-3	1101
6	std. err >3	1010
7	std. err <- 3	1110
8	error	1011
9	null	1111

for m = 20, n = 2, k = 18, and r = 1, the above scheme's worst case compression ratio is same as before

$$100 \times (1 - (2 \times 4 + 18/2 + 4)/(21 \times 4)) = 75.0\%$$

But, for m = 20, n = 2, k = 18, and r = 1, the above scheme provides best case compression ratio of

$$100 \times (1 - (2 \times 4 \times 8 + 18 + 4 \times 8)/(21 \times 4 \times 8)) = 83.04\%,$$

and for m = 40, n = 2, k = 38, and r = 1, the above scheme provides best case compression ratio of

$$100 \times (1 - (2 \times 4 \times 8 + 38 + 4 \times 8)/(41 \times 4 \times 8)) = 89.79\%$$

Instead of compressing machine data, the above algorithm may be used to increase precision for the range of data that occurs more frequently and decrease precision for the data that happens infrequently. For example, additional bits can be assigned to represent data that have standard errors in the range ± 3 *z*-scores and fewer bits for data that have standard error outside of that range.

In some embodiments, the presently disclosed technology does not involve sending at least some part of the actual data; rather, the technology uses parameters and/or statistical errors to implicitly communicate the actual data. Therefore, the system may be used as a data obfuscation technique. In some embodiments, the actual, exact data values cannot be recovered from the sensor standard error values without prior knowledge of the model and model parameters. If the model is encrypted during transmission, only the independent variables need be sent encrypted during transmission. The standard errors for dependent sensor variables may be sent as plain text, thus reducing the transmission encryption overhead and improving performance.

A linear model generated by machine learning may also be used, which greatly decreases the model size as compared to other modeling techniques. Since only two model parameters are required (*i.e.*, offset and gradient) and relatively little computing resources are needed to generate a linear model, the recalculation and re-transmission of the model can occur more frequently and on any transmission interface, *e.g.*, on satellite, LoRaWAN, cellular, etc. Additionally, range-based linear models may also be used. For example, the full operating range of independent parameters are divided into ‘n’ smaller ranges and ‘n’ linear models are computed for each smaller range. Considering that only a few variables are required to store linear models, the combined model size would remain very small (*e.g.*, 100 range based models require 100 × 2 parameters per model × 4 bytes per parameter + 100 × (1 error mean + 1 error standard deviations) × 4 bytes each = 1600 bytes or 4 orders of magnitude smaller than the model lookup table referenced above).

Figure 1 shows a block diagram of a system including at least a transmitter 102 and a receiver 112 according to some embodiments of the presently disclosed technology. As described above, the transmitter 102 can be an edge device that receives a data stream 108 from one or more sensors. Some embodiments of an edge device are described in U.S. Application No. 15/703,487 filed September 13, 2017. The transmitter 102 can include one or more processors 104 configured to implement statistical model(s) 106 and encode data differentials (e.g., into bits representation) with respect to the statistical model(s) used. The transmitter 102 is communicatively connected to the receiver 112 via network connection(s) 120. The receiver 112 can be a server including one or more processors 114 configured to implement statistical model(s) 116 and decode data differentials (e.g., from bits representation) with respect to the statistical model(s) used. After or as the decoding is performed, the receiver 112 can generate a reconstructed data stream 118 and provide it to another device or a user.

As an example, the transmitter may be constructed as follows. A controller of the transmitter may include any or any combination of a system-on-chip, or commercially available embedded processor, Arduino, MeOS, MicroPython, Raspberry Pi, or other type processor board. The transmitter may also include an Application Specific Integrated Circuit (ASIC), an electronic circuit, a programmable combinatorial circuit (*e.g.*, FPGA), a processor (shared, dedicated, or group) or memory (shared, dedicated, or group) that may execute one or more software or firmware programs, or other suitable components that provide the described functionality. The controller has an interface to a communication port, *e.g.*, a radio or network device.

In embodiments, one or more of sensors determine, sense, and/or provide to controller data regarding one or more other characteristics may be and/or include Internet of Things ("IoT") devices. IoT devices may be objects or "things", each of which may be embedded with hardware or software that may enable connectivity to a network, typically to provide information to a system, such as controller. Because the IoT devices are enabled to communicate over a network, the IoT devices may exchange event-based data with service providers or systems in order to enhance or complement the services that may be provided. These IoT devices are typically able to transmit data autonomously or with little to no user intervention. In embodiments, a connection may accommodate vehicle sensors as IoT devices and may include IoT-compatible connectivity, which may include any or all of WiFi, LoRan, 900 MHz Wifi, BlueTooth, low-energy BlueTooth, USB, UWB, etc. Wired connections, such as Ethernet 1000baseT, CANBus, USB 3.0, USB 3.1, etc., may be employed.

Embodiments may be implemented into a computing device or system using any suitable hardware and/or software to configure as desired. The computing device may house a board such as motherboard which may include a number of components, including but not limited to a processor and at least one communication interface device. The processor may include one or more processor cores physically and electrically coupled to the motherboard. The at least one communication interface device may also be physically and electrically coupled to the motherboard. In further implementations, the communication interface device may be part of the processor. In embodiments, processor may include a hardware accelerator (*e.g.*, FPGA).

Depending on its applications, computing device may include other components which include, but are not limited to, volatile memory (*e.g.*, DRAM), non-volatile memory (*e.g.*, ROM), and flash memory. In embodiments, flash and/or ROM may include executable programming instructions configured to implement the algorithms, operating system, applications, user interface, etc.

In embodiments, computing device may further include an analog-to-digital converter, a digital-to-analog converter, a programmable gain amplifier, a sample-and-hold amplifier, a data acquisition subsystem, a pulse width modulator input, a pulse width modulator output, a graphics processor, a digital signal processor, a crypto processor, a chipset, a cellular radio, an antenna, a display, a touchscreen display, a touchscreen controller, a battery, an audio codec, a video codec, a power amplifier, a global positioning system (GPS) device or subsystem, a compass (magnetometer), an accelerometer, a barometer (manometer), a gyroscope, a speaker, a camera, a mass storage device (such as a SIM card interface, and SD memory or micro-SD memory interface, SATA interface, hard disk drive, compact disk (CD), digital versatile disk (DVD), and so forth), a microphone, a filter, an oscillator, a pressure sensor, and/or an RFID chip.

The communication network interface device may enable wireless communications for the transfer of data to and from the computing device. The term “wireless” and its derivatives may be used to describe circuits, devices, systems, processes, techniques, communications channels, etc., that may communicate data through the use of modulated electromagnetic radiation through a non-solid medium. The term does not imply that the associated devices do not contain any wires, although in some embodiments they might not. The communication chip 406 may implement any of a number of wireless standards or protocols, including but not limited to Institute for Electrical and Electronic Engineers (IEEE) standards including Wi-Fi (IEEE 802.11 family), IEEE 802.16 standards (*e.g.*, IEEE 802.16-2005 Amendment), Long-Term Evolution (LTE) project along with any amendments, updates, and/or revisions (*e.g.*, advanced LTE project, ultra-mobile broadband (UMB) project (also referred to as “3GPP2”), etc.). IEEE 802.16 compatible BWA networks are generally referred to as WiMAX networks, an acronym that stands for Worldwide Interoperability for Microwave Access, which is a certification mark for products that pass conformity and interoperability tests for the IEEE 802.16 standards. The communication chip 406 may operate in accordance with a Global System for Mobile Communication (GSM), General Packet Radio Service (GPRS), Universal Mobile Telecommunications System (UMTS), High Speed Packet Access (HSPA), Evolved HSPA (E-HSPA), or LTE network. The communication chip 406 may operate in accordance with Enhanced Data for GSM Evolution (EDGE), GSM EDGE Radio Access Network (GERAN), Universal Terrestrial Radio Access Network (UTRAN), or Evolved UTRAN (E-UTRAN). The communication chip 406 may operate in accordance with Code Division Multiple Access (CDMA), Time Division Multiple Access (TDMA), Digital Enhanced Cordless Telecommunications (DECT), Evolution-Data Optimized (EV-DO), derivatives thereof, as well as any other wireless protocols that are designated as 3G, 4G, 5G, and beyond. The communication chip may operate in accordance with other wireless protocols in other embodiments. The computing device may include a plurality of communication chips. For instance, a first communication chip may be dedicated to shorter range wireless communications such as Wi-Fi and Bluetooth and a second communication chip may be dedicated to longer range wireless communications such as GPS, EDGE, GPRS, CDMA, WiMAX, LTE, Ev-DO, and others.

The processor of the computing device may include a die in a package assembly. The term “processor” may refer to any device or portion of a device that processes electronic data from registers and/or memory to transform that electronic data into other electronic data that may be stored in registers and/or memory.

Figure 2 shows a flowchart of actions performed by the transmitter and the receiver of Figure 1, according to some embodiments of the presently disclose technology.

As shown in Figure 2, at block 202 the transmitter receives a stream of sensor data. At block 204, the transmitter receives inputs on or automatically generates statistical model(s) that describe or otherwise model the stream of sensor data. At block 206, the transmitter transmits to the receiver the statistical model(s) or data (e.g., model parameters) defining the statistical model(s). At block 208, the transmitter receives subsequent sensor data of the stream. At block 210, the transmitter calculates a difference between the subsequent sensor data and the expectation (e.g., predicted values) based on the statistical model(s). At block 212, the transmitter encodes the difference data (e.g., into bits representations). At block 214, the transmitter transmits the encoded difference data to the receiver.

With continued reference to Figure 2, at block 222, the receiver receives from the transmitter the statistical model(s) or the data defining the statistical model(s). At block 224, the receiver receives the encoded difference data. At block 226, the receiver decodes the difference data. At block 228, the receiver uses the statistical model(s) and the decoded difference data to estimate the subsequent sensor data. At block 230, the receiver outputs the estimated subsequent sensor data.

Although certain embodiments have been illustrated and described herein for purposes of description, a wide variety of alternate and/or equivalent embodiments or implementations calculated to achieve the same purposes may be substituted for the embodiments shown and described without departing from the scope of the present disclosure. The various embodiments and optional features recited herein may be employed in any combination, subcombination, or permutation, consistent with the discussions herein. This application is intended to cover any adaptations or variations of the embodiments discussed herein, limited only by the claims.

The various embodiments described above can be combined to provide further embodiments. All of the U.S. patents, U.S. patent application publications, U.S. patent applications, foreign patents, foreign patent applications and non-patent publications referred to in this specification and/or listed in the Application Data Sheet are incorporated herein by reference, in their entirety. Aspects of the embodiments can be modified, if necessary to employ concepts of the various patents, applications and publications to provide yet further embodiments. In cases where any document incorporated by reference conflicts with the present application, the present application controls.

These and other changes can be made to the embodiments in light of the above-detailed description. In general, in the following claims, the terms used should not be construed to limit the claims to the specific embodiments disclosed in the specification and the claims, but should be construed to include all possible embodiments along with the full scope of equivalents to which such claims are entitled. Accordingly, the claims are not limited by the disclosure.

CLAIMS

1. A computer-readable storage medium storing contents that, when executed by one or more processors, cause the one or more processors to perform actions for communicating information between a transmitting device and a receiving device, the actions comprising:

communicating, from the transmitting device to the receiving device, information defining one or more predictive statistical models, wherein the one or more predictive statistical models are configured to predict a second category of sensor data based at least partially on a first category of sensor data; and

after communicating the information defining the one or more predictive statistical models, communicating, from the transmitting device to the receiving device, (a) instances of the first category of sensor data and (b) at least a difference between the instances of the second category of sensor data and a prediction made by the one or more statistical models based on the instances of the first category of sensor data, to convey at least instances of the second category of sensor data to the receiving device in an implicit form.

2. A computer-implemented method of communicating information between a transmitting device and a receiving device, comprising:

obtaining information defining one or more predictive statistical models, wherein the one or more predictive statistical models are configured to predict a second category of sensor data based at least partially on a first category of sensor data; and

communicating, from the transmitting device to the receiving device, (a) instances of the first category of sensor data and (b) error data associated with a prediction by the one or more statistical models, to convey at least instances of the second category of sensor data in an implicit form.

3. The method of claim 2, wherein the first category of sensor data corresponds to independent variables of the one or more predictive statistical models

and the second category of sensor data corresponds to dependent variables of the one or more predictive statistical models.

4. The method of claim 2, wherein the error data associated with a prediction by the one or more statistical models comprises an indication of difference between at least one of the instances of the second category of sensor data and a prediction made by the one or more statistical models based on the instances of the first category of sensor data.

5. The method of claim 2, wherein the one or more predictive statistical models are trained based at least partially on previous instances of the first and second categories of sensor data.

6. The method of claim 2, further comprising updating the one or more predicative statistical models based at least partially on the instances of the first category of sensor data and the instances of the second category of sensor data.

7. A method of communicating information, comprising:

modeling a stream of sensor data, to produce parameters of a predictive statistical model;

communicating information defining the predictive statistical model from a transmitter to a receiver; and

after communicating the information defining the predictive statistical model to the receiver, communicating information characterizing subsequent sensor data from the transmitter to the receiver, dependent on an error of the subsequent sensor data with respect to a prediction of the subsequent sensor data by the statistical model.

8. The method of claim 7, further comprising calculating, at the receiver, the subsequent sensor data based at least partially on the error of the sensor data and the prediction of the sensor data by statistical model.

9. The method of claim 7, further comprising acquiring a time series of subsequent sensor data, and communicating from the transmitter to the receiver, information characterizing the time series of subsequent sensor data comprising a time series of errors of subsequent sensor data time samples with respect to a prediction of the subsequent sensor data time samples by the predictive statistical model.

10. The method of claim 7, wherein the predictive statistical model is adaptive to the communicated information characterizing subsequent sensor data.

11. The method of claim 7, further comprising storing information dependent on the predictive statistical model in a memory of the transmitter and a memory of the receiver.

12. The method of claim 7, further comprising determining a sensor data standard error based on a predicted sensor data error standard deviation.

13. The method of claim 7, wherein the predictive statistical model is derived based at least partially on relationships between independent and dependent variables represented in the sensor data.

14. The method of claim 13, wherein the predictive statistical model generates error statistics comprising a mean training error and a standard deviation of the mean training error for a stream of sensor data of the training data set in a training period.

15. The method of claim 14, further comprising computing a predicted stream of sensor data, a predicted stream of sensor data error means, and a predicted stream of sensor data error standard deviations, based on the predictive statistical model.

16. The method of claim 15, further comprising communicating the predicted stream of sensor data error means from the transmitter to the receiver.

17. The method of claim 16, further comprising receiving the predicted stream of sensor data error means at the receiver, and based on the predictive statistical model and the received stream of sensor data error means, reconstructing the stream of sensor data.

18. A system, comprising:

one or more processors; and

memory storing contents that, when executed by the one or more processors, cause the system to:

obtain information defining a predictive statistical model that models a stream of sensor data; and

communicate information characterizing subsequent sensor data from a transmitter to a receiver, dependent on an error of the subsequent sensor data with respect to a prediction of the subsequent sensor data by the statistical model.

19. The system of claim 18, wherein the contents, when executed by the one or more processors, further cause the system to transmit a standard error of the prediction of the subsequent sensor data by the predictive statistical model from the transmitter to the receiver.

20. The system of claim 18, wherein the stream of sensor data comprises sensor data from a plurality of sensors which are dependent on at least one common control variable, the predictive statistical model being dependent on a correlation of the sensor data from the plurality of sensors, and the contents, when executed by the one or more processors, further cause the system to:

calculate standard errors of the subsequent sensor data from the plurality of sensors with respect to the predictive statistical model dependent on a correlation of the sensor data, entropy encoding the standard errors based on at least the correlation; and

transmit the entropy encoded standard errors, and a representation of the at least one common control variable from the transmitter to the receiver.

21. The system of claim 18, wherein the stream of sensor data comprises timestamped data comprising at least one of engine speed, engine load, coolant temperature, coolant pressure, oil temperature, oil pressure, fuel pressure, or fuel actuator state.

22. The system of claim 18, wherein the predictive statistical model comprises at least one of a spline model, a neural network, a support vector machine, or a Generalized Additive Model (GAM).

23. The system of claim 18, wherein the stream of sensor data comprises temporally averaged sensor data for a series of timestamps.

24. The system of claim 18, wherein communications between the transmitter to the receiver are bandwidth constrained.

25. The system of claim 18, wherein the transmitter and receiver are asymmetric, wherein the transmitter is a data source and the receiver is a data sink, wherein the receiver is configured to receive communications from a plurality of transmitters.

26. The system of claim 18, wherein the information characterizing subsequent sensor data comprising the error of subsequent sensor data with respect to the prediction of the subsequent sensor data by the predictive statistical model comprises a standardized training error mean, standardized by subtracting a training error mean from an instantaneous error between subsequent sensor data and predicted

subsequent sensor data, and dividing this difference by a training error standard deviation for a respective sensor, to produce a z-score of a prediction error.

27. The system of claim 18, wherein the error of the subsequent sensor data with respect to the prediction of the subsequent sensor data is statistically normalized and quantized with respect to units of standard deviation away from a predicted mean of the subsequent sensor data.

28. The system of claim 18, wherein the error of the subsequent sensor data with respect to the prediction of the subsequent sensor data is quantized in uneven steps with respect to units of standard deviation away from a predicted mean of the subsequent sensor data.

29. The system of claim 18, wherein the error of the subsequent sensor data with respect to the prediction of the subsequent sensor data is represented with higher resolution for smaller deviation away from a predicted mean of the subsequent sensor data than for higher deviation from the predicted mean of the subsequent sensor data.

30. The system of claim 18, wherein the communicating information characterizing subsequent sensor data comprises communicating encrypted information representing independent variables and unencrypted information representing dependent variables.

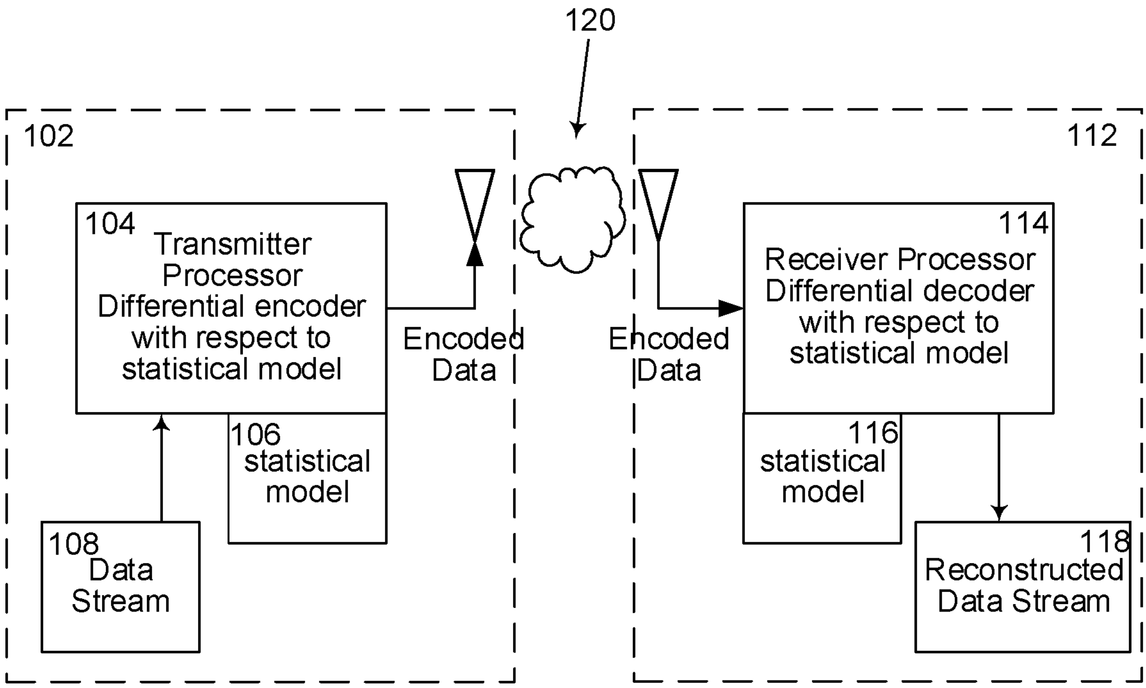
120
102
104
Transmitter Processor
Differential encoder with respect to statistical model
Encoded Data
106
statistical model
108
Data Stream
112
114
Receiver Processor
Differential decoder with respect to statistical model
Encoded Data
116
statistical model
118
Reconstructed Data Stream

FIG. 1

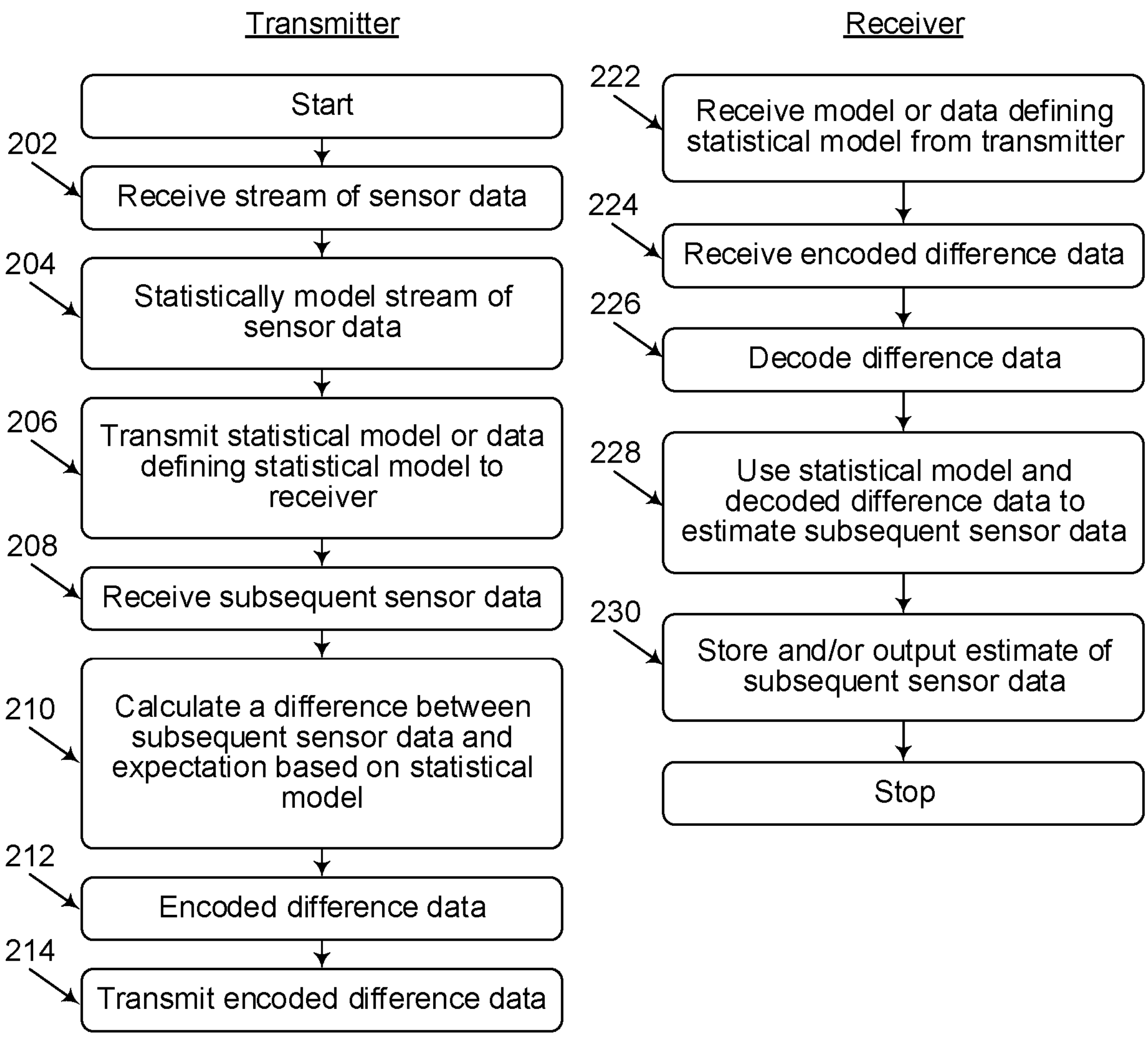
Transmitter
Start
202
Receive stream of sensor data
204
Statistically model stream of sensor data
206
Transmit statistical model or data defining statistical model to receiver
208
Receive subsequent sensor data
210
Calculate a difference between subsequent sensor data and expectation based on statistical model
212
Encoded difference data
214
Transmit encoded difference data
Receiver
222
Receive model or data defining statistical model from transmitter
224
Receive encoded difference data
226
Decode difference data
228
Use statistical model and decoded difference data to estimate subsequent sensor data
230
Store and/or output estimate of subsequent sensor data
Stop

FIG. 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US20/15698

A. CLASSIFICATION OF SUBJECT MATTER

IPC - G06N 5/02; G06Q 10/06; G06F 11/00 (2020.01)

CPC - G06N 5/02; G06Q 10/06; G06N 7/005, 20/00; G06F 11/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y --- A	US 2019/0057170 A1 (ARUNDO ANALYTICS, INC) 21 February 2019; figure 1, paragraphs [0022]-[0023]	1-6 --- 7-13, 18, 19, 21-25, 30 --- 14-17, 20 26-29
Y --- A	US 2014/0351183 A1 (LANDMARK GRAPHICS CORPORATION) 27 November 2014; paragraphs [0057], [0061], [0069], [0138]	7-13, 18, 19, 22-25, 30 --- 1-6, 14-17, 20-21, 26-29
Y	US 2017/0076209 A1 (WELLAWARE HOLDINGS, INC) 16 March 2017; paragraph [0072]	18, 21
A	US 2003/0065409 A1 (RAETH, P et al.) 03 April 2003; entire document	1-30
A	US 2007/0112521 A1 (AKIMOV, O et al.) 17 May 2007; entire document	1-30
A	US 5,682,317 A (KEELER, J et al.) 28 October 1997; entire document	1-30
A	US 6,804,600 B1 (ULUYOL, O et al.) 12 October 2004; entire document	1-30
A	US 2002/0091972 A1 (HARRIS, D et al.) 11 July 2002; entire document	1-30
A	US 2012/0316835 A1 (MAEDA, S et al.) 13 December 2012; entire document	1-30
A	US 2006/0036403 A1 (WEGERICH, S et al.) 16 February 2006; entire document	1-30

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 March 2020 (23.03.2020)	23 APR 2020

Name and mailing address of the ISA/US	Authorized officer
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Shane Thomas Telephone No. PCT Helpdesk: 571-272-4300